ATHENS UNIVERSITY OF ECONOMICS AND BUSSINESS

xSDR – User's Guide

eXtensible Suite for Dimensionality Reduction

DB-Net Research Team

5/5/2010

The present document is a user's manual of the xSDR tool, a database dimensionality reduction and data mining suite. The guide was compiled in the framework of the master thesis of Anastasios Kapernekas, under the supervision of Prof. Michalis Varzygiannis, Athens University of Economics and Business, and Dr. Panagis Magdalinos, Athens University of Economic and Business.

Index

1. System Review 3
1.1 Improvements compared to DRC 4
2. System Requirements 5
3. Application Installation 6
3.1 Prerequisites 6
3.2 Working with the executable file 8
3.3 Working with the source code 8
4. User's Guide 9
4.1 First Layer – Data Input 9
4.1.1 Data input from Relational Database System 12
4.1.2 Data input from test 16
4.1.3 Data input from a stored dataset 21
4.2. Second Level – Data Transformation 24
4.3. Third level – Dimensionality Reduction Algorithms Execution 32
4.3.1. DR Toolbox Algorithm Execution 34
4.3.1.1 Incorporation of new versions of DRToolbox 35
4.3.2 Custom algorithms execution 39
4.3.2.1 Custom Algorithm writing 43
4.3.3 Distributed Algorithms Execution 45
4.4. Fourth Level – Results Visualization 45
4.4.1 "Select all" option 47
4.4.2. Data Points Graph 48
4.4.3. "Comparison" projection 50
4.5. Fifth level - Data mining with Weka tool 50
4.6. Application general settings 52
5. Annex 55
5.1. Links 55

1. System Review

xSDR is a modern and powerful data mining mechanism. The domain of data mining is one of the most popular research domains in information science. The constantly growing memory volumes, combined with the need for data mining in most modern data applications have led to the idea of creating a single modern tool, that will facilitate a fast and effective management of large volume data.

xSDR is the evolution of DRC (Dimensionality Reduction Console) application. Therefore, the system requirements have remained the same. Thus, the application provides the user with the following options:

- **Data origin option**. The user can select the origin of his data among the most popular databases (Oracle, SQL Server, My SQL) or text files (.txt, .dat).
- **Data processing**. The user has the option to filter and process the data, according to his/her wishes: there is a wide range of possibilities, such as replacement of alphanumeric with numeric data, value substitution in numerical data, normalization of values in specific stress or simple data mathematical calculations. In addition, the user has the option to exclude dimensions at will from the dimensionality reduction process.
- **Dimensionality reduction expandable algorithms**. This is the part of the application that was mostly emphasized. The user must have the option to execute every algorithm he wishes to. This way, the option of importing algorithms by the user is necessary. Furthermore, there should not be any limitation in the choice of the algorithm, so the system will have to be able to execute centralized, as well as distributed algorithms. Therefore, xSDR covers the options of both central or distributed algorithm execution. The user can select one of the incorporated algorithms of the program (every application algorithm derive from the drToolbox as in the initial version) or import his own.
- **Results storing**. The whole dimensionality reduction procedure would be meaningless, if the user were not able to store the data for future use. In this case, different alternatives are also offered. The produced data can be saved as Arff files, text files or relational databases.
- **Graphical visualization of results**. The application permits the deduction of useful results through graphic visualization of the dimensionality reduction results.
- **Weka modules execution option**. Weka is a very powerful data mining tool. Its development is constant and the exported results are considerably useful, mainly to research applications. Therefore, since our tool to a large extent addresses students and mining data researchers, we included the integration of the "Weka modules" execution option in the application's environment.
- **Memory management**. Although the constantly growing amounts of memory available assure partially the capability of large volume data processing, we considered necessary the addition of a control mechanism to the program. As a result, our application controls the available memory amounts and warns the user in case the memory is not sufficient for loading a certain database.

1.1 Improvements compared to DRC

As mentioned before, xSDR is the evolution of DRC. The graphic environment has remained unchanged, so that it will not create problems to older users. Simultaneously, the following improvements and additions took place:

- Compatibility problems with newer versions of Matlab (DRC could only be executed by using the Compiler of Matlab 2007b edition) have been resolved
- Application development for correct execution in Windows 7 environment
- Connection completion with COM Automation Servers for the execution of custom algorithms.
- Input and export data modification for users' algorithms. Nevertheless, the XML header of the files has remained unchanged.
- Integration of renewed algorithms from DRToolbox.
- Addition of stored database sources deletion option.
- Addition of stored database sources settings retaining option or new settings introduction
- Abolition of Microsoft Access input option
- Selection of any dimension as data class
- Correction of problematic Weka modules during the execution.

2. System Requirements

xSDR application is designed so that it does not have high hardware requirements. The only major hardware requirement concerns memory size. Available memory should be large enough, so as to load every chosen dataset. The program manages the available memory and if the size of chosen dataset exceeds the system memory, a warning message appears to avoid memory overload.

As far as software is concerned, there are a number of requirements, as well as limitations. Our application is written in C# and its execution necessitates the installation of Microsoft .Net Framework 3.5 is required. Due to the latter, xSDR, cannot be executed in any OS environment other than Windows. xSDR has been tested in Windows XP, Windows Vista & Windows 7 environment, whereas it is not compatible with older Windows versions, for which .Net Framework is not available.

Consequently, the limitations we are setting software-wise are those of .Net Framework with SP1 and of MS Charts Control, items necessary to the execution.

System requirements

- **Supported operational systems:** Windows Vista, Windows XP SP 3, Windows 7
- **.NET Framework:** .NET Framework 3.5 SP1
- **Processor:** 400 MHz Pentium or equivalent (minimum requirement), 1GHz Pentium or equivalent (recommended)
- **RAM:** 96 MB (minimum), 256 MB (recommended)
- **Hard Disk:** 500 MB of free space
- **CD/DVD Driver:** not required
- **Display:** 800 x 600, 256 colors (minimum), 1024 x 768 high color, 32 bit (recommended)

The second limitation concerns the execution of dimensionality reduction algorithms. All dimensionality reduction algorithms (both those included in the program and those that are imported by the user) are written in Matlab language. Due to the latter, the execution of these algorithms presupposes the installation of Matlab compiler. Thus, the user has to install the whole Matlab suite, or MCR (Matlab Compiler Runtime), which is available for free installation by Mathworks website (the Matlab developing company). At this point, we have to underline that there are some limitations regarding Matlab and MCR edition that are compatible with the application. These limitations are clearly mentioned in chapter 3, where the installation procedure is described.

Finally, for the visualization of the dimensionality reduction graphic results, Microsoft Chart Control library is required, which is offered for free by Microsoft website. The necessary links for the application execution can be found in the annex of this guide.

3. Application Installation

There are actually two different ways for someone to work with our tool. The first one is to install the final version (1.0.0) on his machine. This can be done by downloading the appropriate compressed file from the tool's web site and following the instructions provided in the next chapter. The second one is to download the tool's source, load the solution on his preferred IDE - Microsoft Visual Studio is the suggested one, since the whole tool is developed by using it, which entails that the solution and the project files are ready for it - build the solution and then run the tool from the VS runtime environment. The advantages of each method are obvious. By choosing the first method everyone can start using the tool, following some very simple steps and without any programming knowledge needed. By choosing the second method (that requires at least a basic knowledge of C# and the use of Visual Studio) the user is able to see in depth the execution of a dimensionality reduction algorithm. He can also make changes in the code and convert the tool to meet his own needs. Further in our analysis, we are going to describe each method in detail.

3.1 Prerequisites

The first thing that the user has to do is to ensure that his machine meets the hardware requirements we described in chapter 1. After that he has to confirm that the following software is installed on the system.

- .Net Framework 3.5
  In case the user's system doesn't have the specified framework installed, he should follow the link specified in chapter 5.1 and download the framework's installation package from Microsoft's web site. Then, install it and restart the machine to complete installation process.

- Microsoft MS Charts Library
  What the user did earlier with the .Net framework, must repeat for the MS Charts library. If the library is not installed on the system, follow the links provided in chapter 5.1 and download the installation package. Then install it on the system and restart the machine to complete installation.

- Matlab / Matlab Compiler Runtime
  xSDR tool requires either Matlab Suite, or Matlab Compiler Runtime to be installed on the system. Let's see each case separetely.
  - If Matlab is installed on the system the first thing the user has to do is to check the version installed. This is necessary, because the DRToolbox assembly file (drtoolbox.dll) has to be built with the same version installed on the system. The provided library is built with Matlab R2009b version. So, if this is the version the user has already installed on your system, you don't have to install anything else.

- If the user has installed a different version of Matlab than R2009b, then there are two options. The first one is to build a new version of DRtoolbox assembly file and embed it to the tool. This process is described in paragraph 4.3.1.1.
  Note: The use of the term "new version" for the drtoolbox assembly file may not be the most suitable, because the version may not be newer than the one provided. It just has to be built with Matlab deployment tool of the same version, as the one installed on our system. This Matlab version can be older than R2009b, or even the Drtoolbox itself can be older than the one provided.
- In case the user doesn't want to build a new assembly file, he can install the provided MCR. Thus xSDR tool will be able to use the right Matlab Compiler Runtime and there is no need to build a new assembly or to change the Matlab version installed on the system.
- In the last case, that the user's system does not have any version of Matlba, then the user can just install the provided Matlab Compiler Runtime and you are ready to run the tool!

- Last thing you have to do is to add Matlab Compiler's Runtime path at Windows System Path Variable. The process depends on the installed operating system and is described in detail below.

Windows Vista - 7

- Click "Start"
- Click "Control Panel"
- Click "System and Security"
- Click "System"
- Click "Advanced System Settings" (may need administrative privileges)
- Select "Advanced" tab
- Click "Environment Variables"
- Under "System Variables" area, scroll down until you see "Path"
- Select "Path" (single click on it) and press "Edit"
- If you have NOT Matlab installed on your system add "C:\Program Files\MATLAB\MATLAB Compiler Runtime\v711\runtime\win32".
  If you have Matlab installed verify that your version's path is already added. You should see something like this: "C:\Program Files\MATLAB\R2009b\runtime\win32; C:\Program Files\MATLAB\R2009b\bin;" depending on the version installed.

Windows XP

- Right Click "My Computer"
- Click "Properties"
- In the System Properties window select "Advanced" tab
- Click "Environment Variables"
- Under "System Variables" area, scroll down until you see "Path"

- Select "Path" (single click on it) and press "Edit"
- If you have NOT Matlab installed on your system add "C:\Program Files\MATLAB\MATLAB Compiler Runtime\v711\runtime\win32".
  If you have Matlab installed, verify that your version's path is already added. You should see something like this: "C:\Program Files\MATLAB\R2009b\runtime\win32; C:\Program Files\MATLAB\R2009b\bin;" depending on the version installed.

3.1 Working with the executable file

Execution of xSDR with the use of the .exe file is simple. The only thing you have to do is to download the provided file from xSDR site and paste it somewhere on our file system. Then you just execute xSDR.exe and you can start using the tool.

We advice you to copy the application folder inside “Program Files” and create a shortcut on the desktop to access the executable file. This way you will reduce the possibility of accidentally deleting any useful files. By default the tool is configured to look for its core files into the folders specified on the settings tab. If the user doesn’t want to use the default paths, he/she has the option to change them and set the desired paths.

Note: If you use the tool under windows vista or windows 7 it is possible that the system will ask for administrative privileges. In this case, you just have to right click the shortcut or the executable file and select “Run as Administrator”.

3.2 Working with the source code

In order to work with source code, you will need Microsoft Visual Studio. A link to the Microsoft Visual Studio Express version which is available for free on the website of Microsoft is given in the annex of this guide. The application development was completed using the 2008 version, so it is essential to have this or any later version.

Since you have installed Visual Studio, you double click on the solution file - xSDR.sln. This will load the whole project implementation. If the version of Visual Studio is newer than the 2008, it will automatically convert the project. In this case, we recommend that you create backup of the application before the conversion (you will be given this opportunity through the dialog of Visual Studio). After loading is completed, you can compile and then build the project. Then you can execute the application, while watching the internal processes within the tool "Debugger" of Visual Studio, which is outside the scope of this guide.

4. User's Guide

The xSDR application is divided in five levels. The first is the data input level, the second is the data transformation, the third – which is essentially the core of our system – is the dimensionality reduction, the fourth is the results' visualization through graphs and schemas, whereas the fifth and last level contains the interaction with the Weka data mining tool.

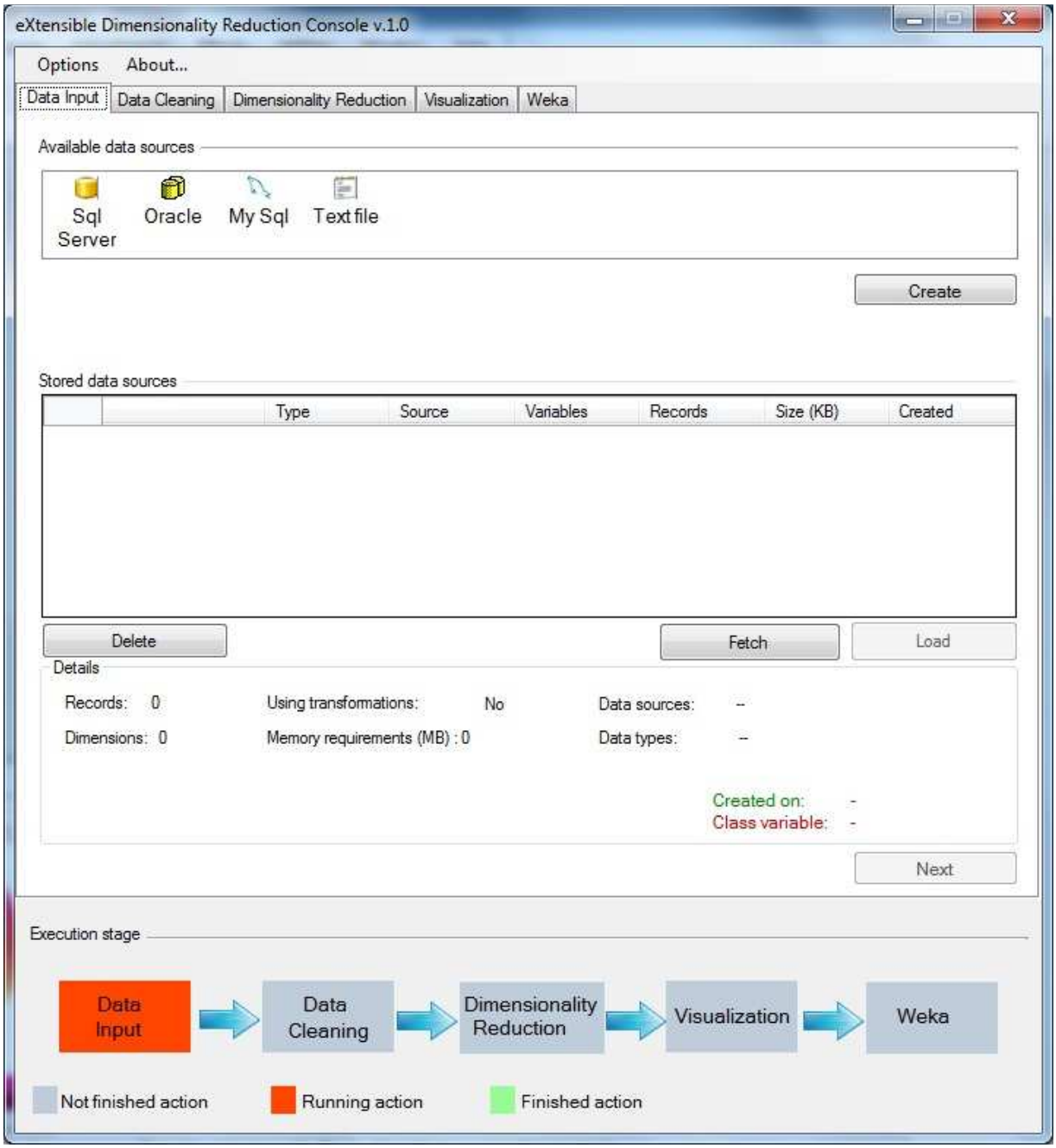
During the whole application design and realization period, user-friendliness and simplicity were given priority, without affecting the application capabilities. We made sure that the user has all configuration options which were set initially in a simple work environment.

Our goal from the beginning was to create a powerful tool that allows students, researchers and everyone else interested in the wider data mining sector to experiment and to create his own dimensionality reduction algorithms in a friendly working environment.

4.1 First Layer – Data Input

The application allows the user to input data into the system by using either some of the best-known database systems (Microsoft SQL Server, Oracle, My Sql) or a text file(.txt, .dat). At the same time, the user can store one or some of his databases, so that he can work on them later in the future. In addition, the user can use the same data configurations or create new ones for his stored database source.

In figure 4.1.1 you can see the application welcome screen. There are six areas, which we are going to analyze below. Use of xSDR at a computer with a minimum resolution of 800 pixels height is highly recommended.



Picture 4.1.1 – The application's main screen

The first area contains the application option menu. Through this menu the user can select the basic program execution parameters. The settings that can be adjusted by the user will be presented in detail in unit 4.6.

The second area consists of cards that correspond to the program execution levels. The user can scroll through all levels and change every previous setting or option. In this case, however, he will have to continue the program execution by the point that the change took place.

In the third area the user can create a new database source for xSDR. In the frame you can see all available data input methods. The user creates a new database source either by double clicking on the correspondent icon in the frame or by choosing the input method and clicking on “Create” button. If he selects a certain database, he will pass to the source connection settings screen, which will be described in unit 4.1.1. If he selects a text file as input, then he arrives at the text file settings screen, which is described in unit 4.1.2.

In the fourth area you can select a previously stored database source to work with. During the program start this list is empty. The stored database sources will appear as soon as you press on “Fetch” button. At this point, we have to note that if you select afterwards a data input from the new database source creation area, this list will disappear. In order to restore it, you press again the “Fetch” button.

For each of the stored databases the following information will be displayed: firstly, for a better optical visualization, you can see the icon that corresponds to the input source. In the next field, you can see the verbal description of the input type and, after that, the name of the source, the columns included in our initial data, the total entries number, the database size in KB and finally the date and the time the source was created. We have to note at this point that the date and the modification time displayed correspond to the date and time you created the database source of xSDR program and not the modification or creation date of the database source.

Consequently, if the user selects one of the stored database sources, three options are available: to delete the stored source, to load it with the configurations and the transformations made during the storage or to load it to its initial form and make the settings and the transformations from the beginning.

To delete a stored database source, the user must just press on the “Delete” button, which is located on the lower part of the stored database source option area.

To load a database source with its stored settings, the user has to press on the “Load” button, having chosen the database source. When the user presses the “Load” button, the option to proceed to the next stage of the application (“Next” button) is activated. In this case, when the user proceeds to the level of data clearance, (the clearance screen is described in unit 4.2), he will realize that the settings and the transformations he had made during the initial database source storage are already selected. Of course their processing is possible.

To revert a database source to its previous state and make the transformations from the beginning, one should double-click on the desired database source through the list of stored sources. In this case – depending on the source input type that you chose – you will be led either to

the connection with database sources settings screen (analytically described in unit 4.1.1) or to the data input from text file screen (analytically described in unit 4.1.2).

As soon as the user has successfully selected the database source, the "proceed" option is enabled. The next level is initiated by pressing the "Next" button.

The fifth area of the welcome screen contains information about the selected database source. In this frame the user can find the following information on the selected input:

- Entries: the number of entries included in the selected database source
- Dimensions: the numbers of columns included in the selected database source
- Transformations: this field informs us, whether the selected database source contains transformations or not
- Required memory: it mentions the memory size in MB that will be used by the selected database source
- Database source: the name of the selected database source (if it is a text file, it is the name of the text file, if it is a database, it is the name of the database)
- Data type: the database source type (SQL Server, My Sql, Oracle, Text File)
- Class: the name of the dimensionality the data class consists of.

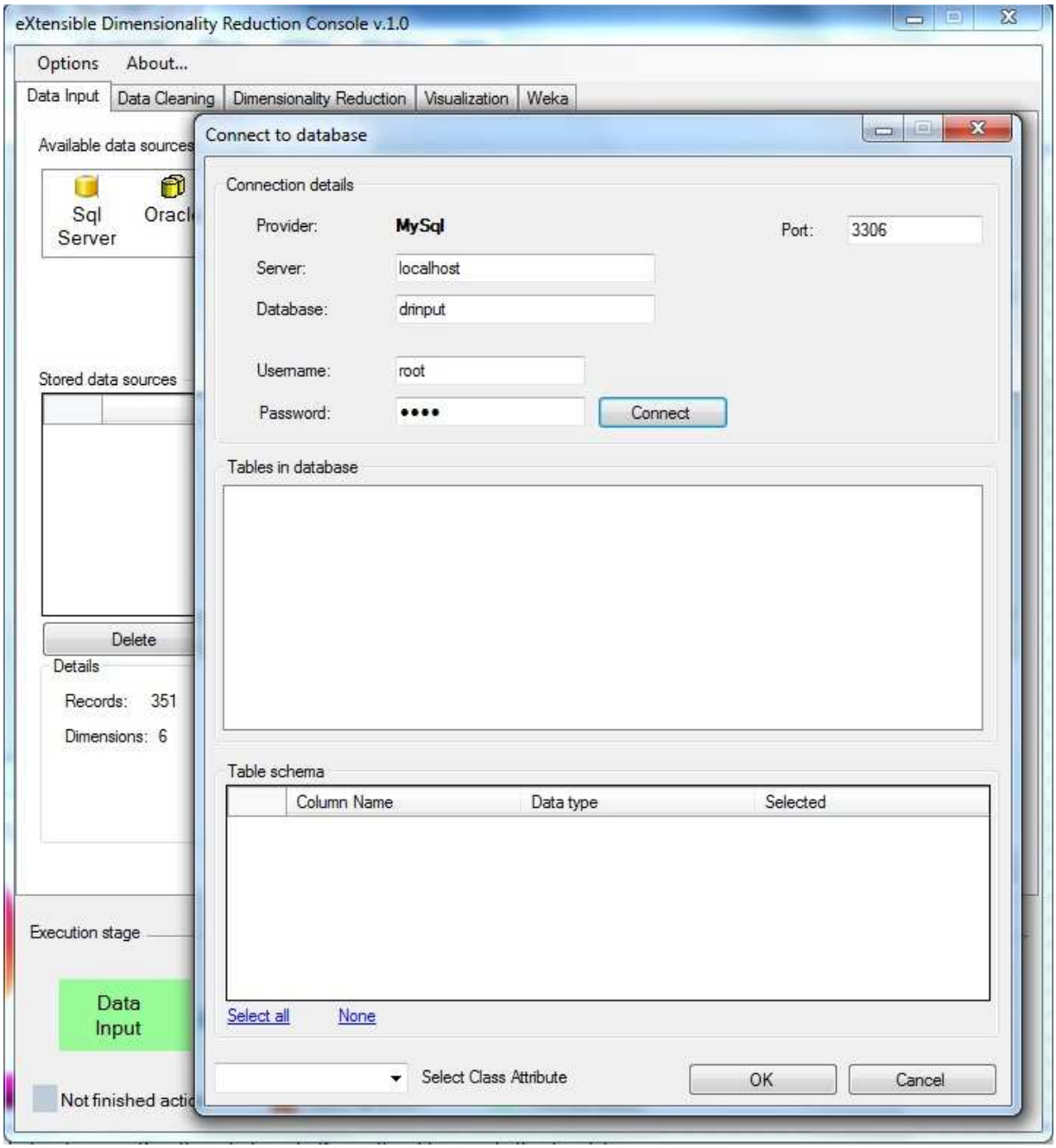
Finally, on the lower part of the screen you can see the application progress bar, on which the five stages of the application appear. The current stage appears in orange, the stages that have not been completed in grey, whereas the stages that have been successfully completed in green.

4.1.1 Data input from Relational Database System

If the user selects as data input a relational database, the connection in database source screen will appear (Picture 4.1.1.1). This screen appears to the user if he chooses to create a new database source from a relational database, or, if he chooses to open a stored database source in its initial form, without the stored settings.

The connection with the database is simple enough. The connection ports are stored (they can change from the program's general settings menu, which we will analyze in unit 4.6), so the user has to input the database provider and the database name he wishes to load. He must also type the required credentials (user name, password).

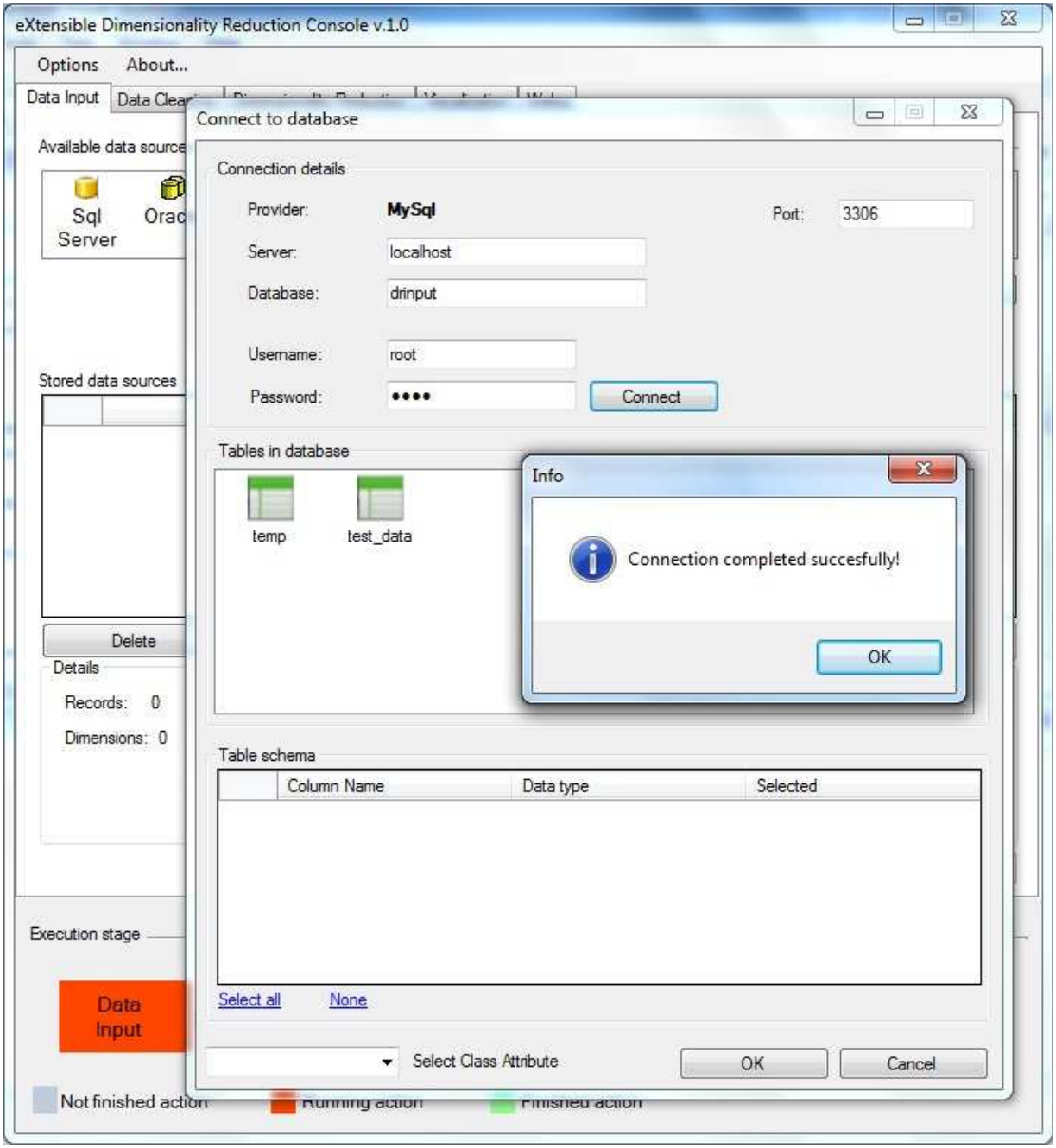
If the information typed by the user is correct, a confirmation message will appear (picture 4.1.1.2) and the connection with the database will be completed. If not, an error message will appear and the user will be asked to input the data again.



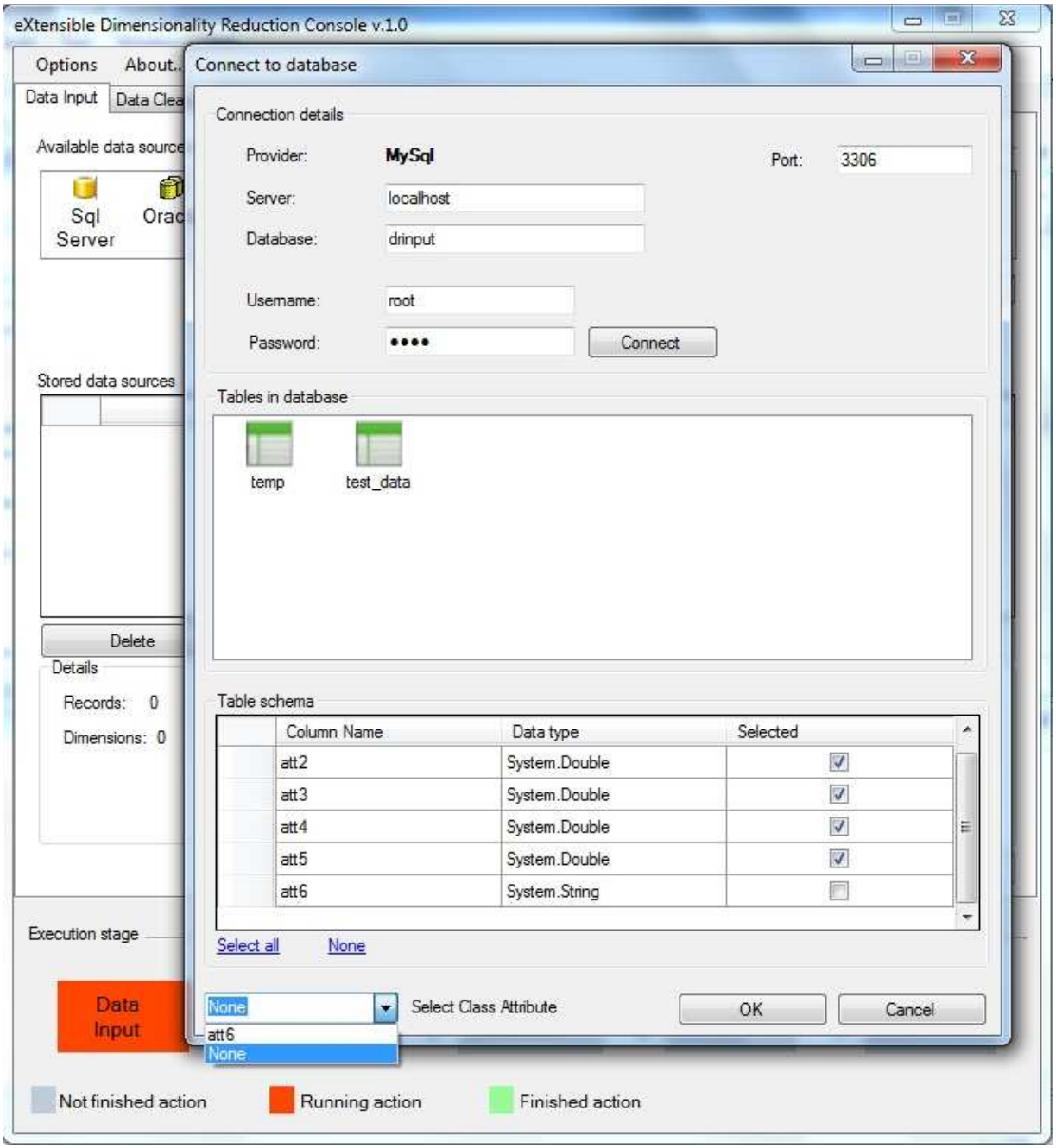
Picture 4.1.1.1 – RDBMS connection form

As soon as the connection to the database is completed, the user can see in the frame the available database tables. Pressing on each one of these tables, the available columns are loaded. For each column the user can see its name and its data type. Finally, there is one more column with a select button. Through this button the user chooses if this column will be used by the program in the next stages of the application. In order to facilitate the user, there are the “Select All” and “Unselect all” options, with which he can select or unselect every database column. Finally, on the lower left part of

the screen there is a drop down menu, through which the user selects the column that forms the dataset class.



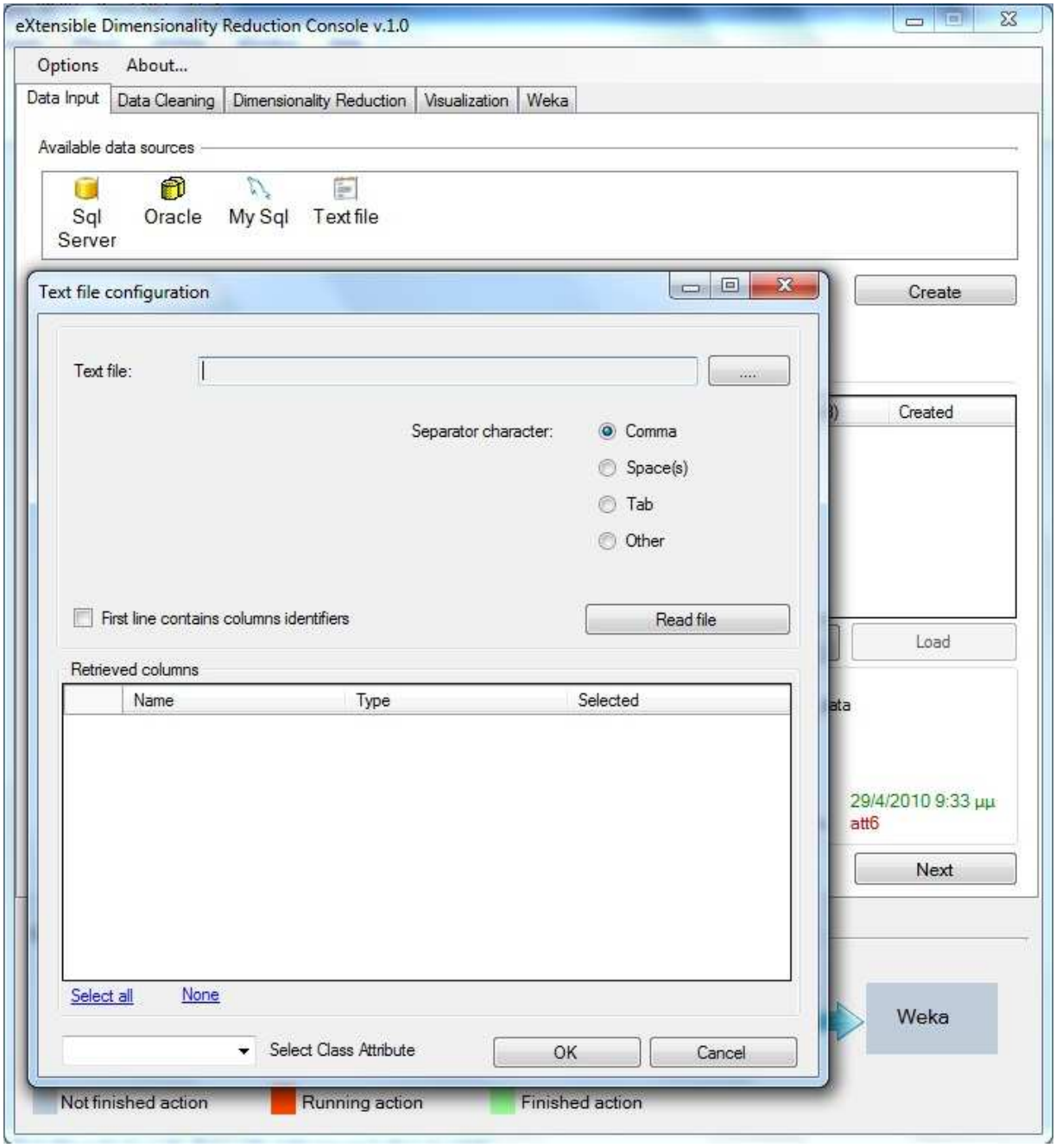
Picture 4.1.1.2 – Successful connection to a RDBMS



Picture 4.1.1.3 – Data select from relational database

4.1.2 Data input from text file

In this paragraph we will describe the text file as database selection procedure. When the user selects a text file as input, either as a new database, or by selecting a stored source without its stored setting, the connection with text file screen will appear (picture 4.1.2.1).



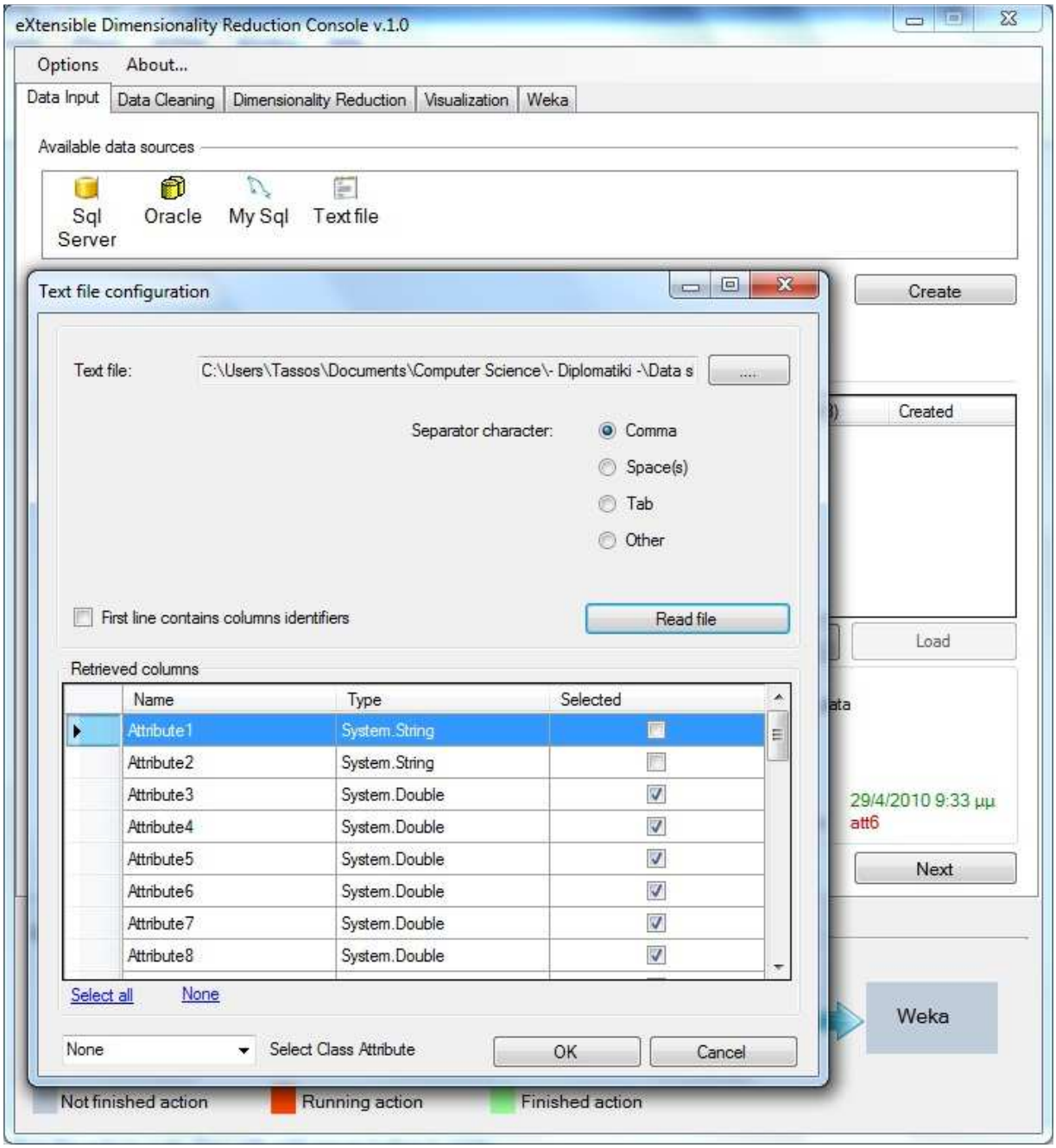
Picture 4.1.2.1 – Connection screen to text file

On the upper part of the screen the user inputs the path for the text file. This can be done either by typing the path or by pressing the “Browse” button, which will open a browsing window, through which the user can navigate until the desired text file. The user can then select a column separator, out of one of the following “radio buttons”. The separator can be space bar, tab, comma or another character which the user will type. In addition, the user can choose if the first line contains the column names. At this point we have to stress that, in the present version; xSDR supports only files that contain column names on the first line.

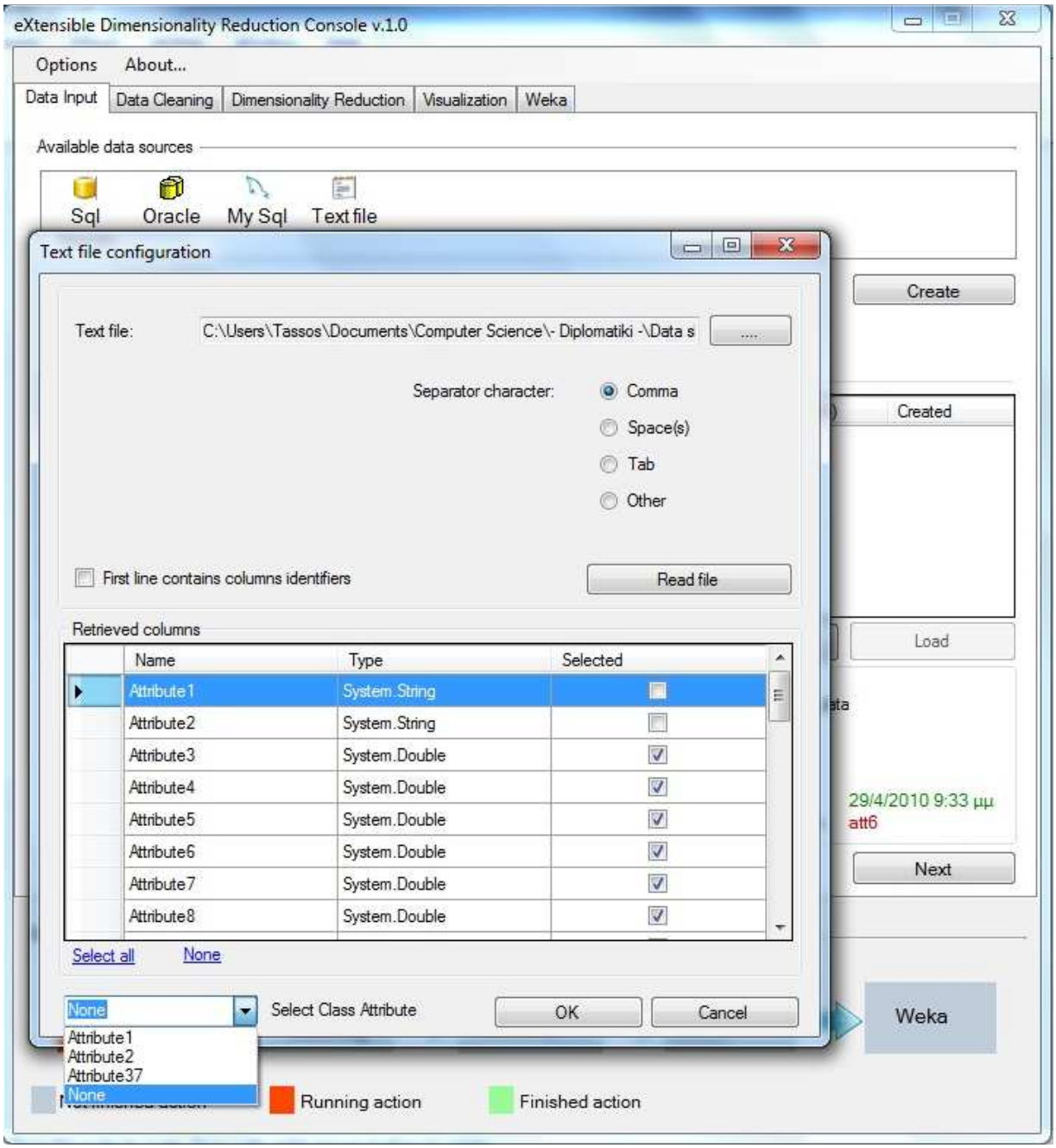
When all these fields are filled, the user can press the file reading button. If the information imported is correct, the available columns of the file imported will appear in the following frame, as seen in figure 4.1.2.2.

After having successfully completed the file reading, you will be able to see in the following frame the available columns (dimensions) of the selected data file. As in the case of the relational database, the column name, the data type and the dimension selection button are displayed. In addition, on the lower left part there is again the data class drop down menu. You can select the data that is going to be used by the program in the following stages. It should be noted here that all dimensions that are not numerical data are deselected by default. When the user selects a dimension as data class, then it is included automatically in the data-set. The rest of the non-numerical dimensions must be selected manually. These functions can be seen in pictures 4.1.2.3 and 4.1.2.4.

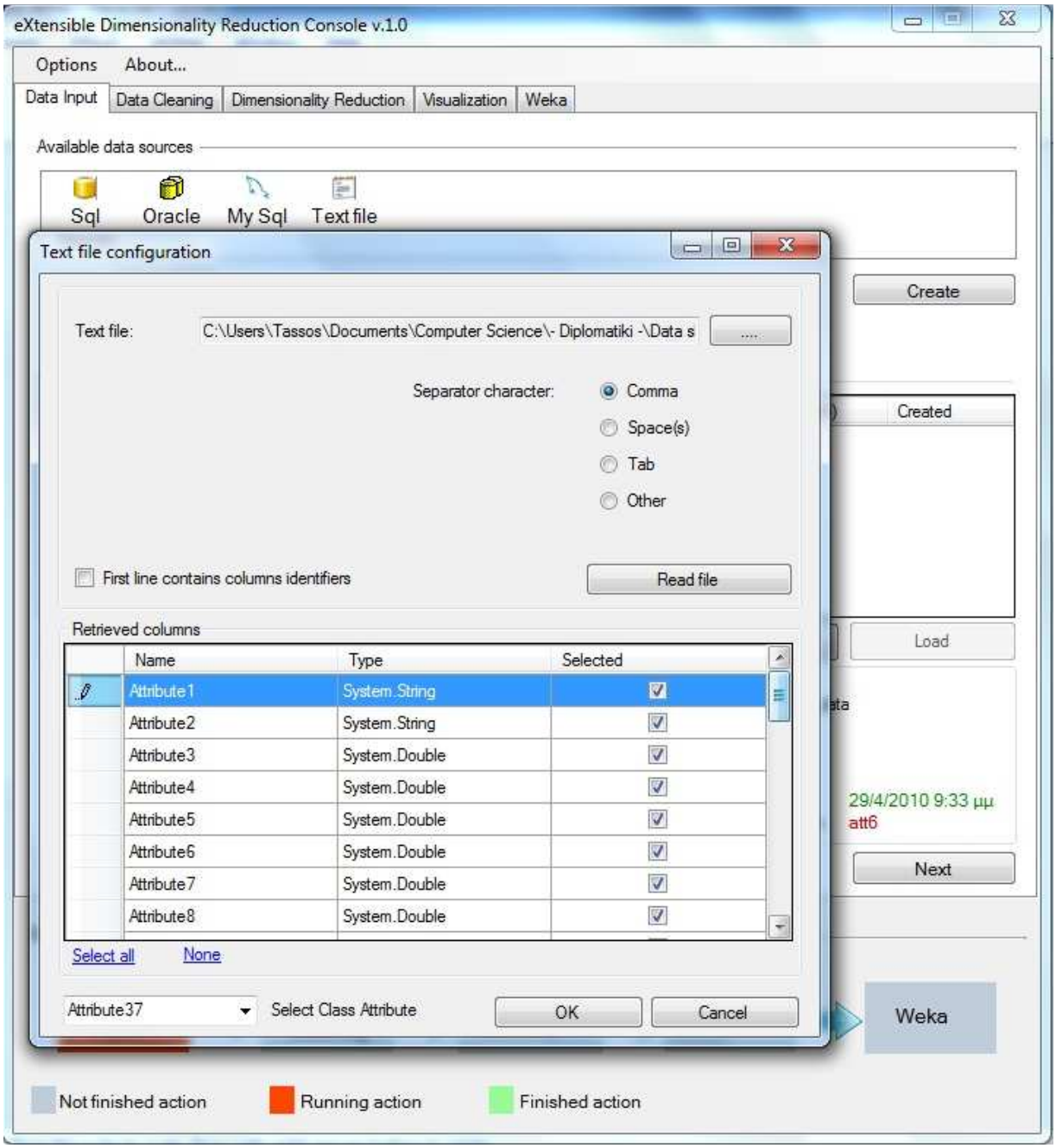
After having finished with data selection, you are ready to press the “OK” button and to close this dialogue window. Of course, there is the possibility to close the dialogue window by cancelling the procedure, without loading the data on memory by pressing the “Cancel” button.



Picture 4.1.2.2 – successful text file read



Picture 4.1.2.3- data class selection



Picture 4.1.2.4 – manual selection of non-arithmetic data

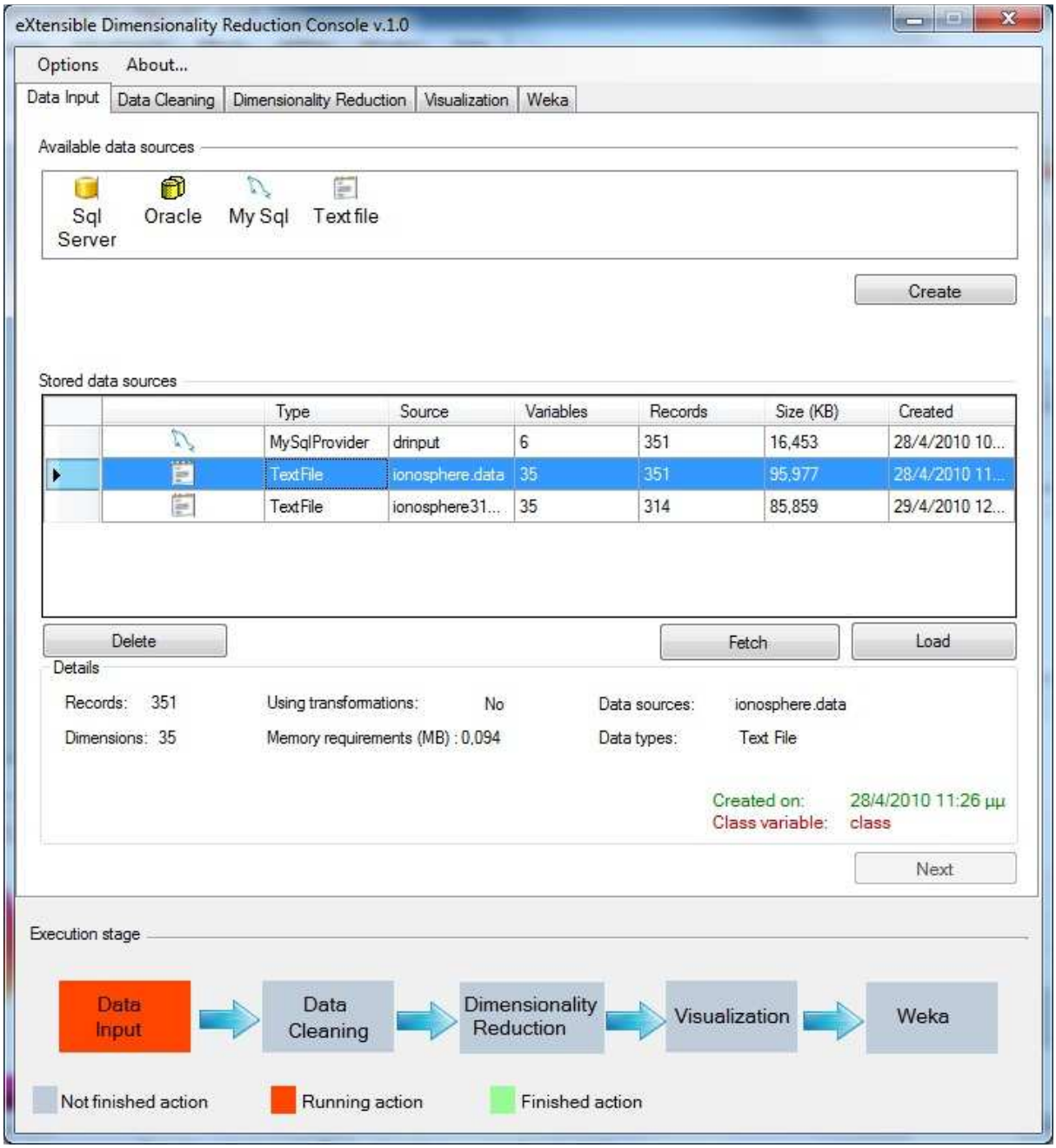
4.1.3 Data input from a stored dataset

As it has already been mentioned, the xSDR suite offers the user the opportunity to store a dataset. Thus, you can configure a dataset as you wish and then work directly with it without re-configuring it. After a dataset has been stored – as described in section 4.2 – you can select it, as you start the tool.

The main screen of the application is shown in Figure 4.1.1. As you can see, in the middle of the screen there is the “Stored Data Sources” grid. At this list the user is able to view all stored datasets. On launching the tool, this list is empty, even if there are some stored datasets from previous executions. What you have to do, in order to access the stored datasets is to press the button “Fetch”. By pressing this button all stored datasets become visible. For ease of selection, you can also have a look at some extra details of the whole dataset, such as the source type (relational database, text file), the name of all the variables contained in the original set, the records, the size and the date saved.

To work with the data set you have selected from the list, you have to pick one of the following scenarios: first one is to use the selection and configuration (data, transformations) carried out, while storing; or you can load the original data set and make a new configuration. In the first case, you just have to press the button "Load", bottom right of the list of saved sets, and all configurations areloaded. If you want to work with the original data set, just double click on the list. Then a new dataset configuration screen will appear and you can make a new configuration.

Finally, the user is able to remove a stored data set from memory, if it is no longer needed. This can be done by pressing the "delete" button on the left of the screen, just below the list of stored data sources. In this case, you will see a dialog box that requires confirmation of the cancellation, as shown in Figure 4.1.3.3



Picture 4.1.3.1 – Stored Datasets

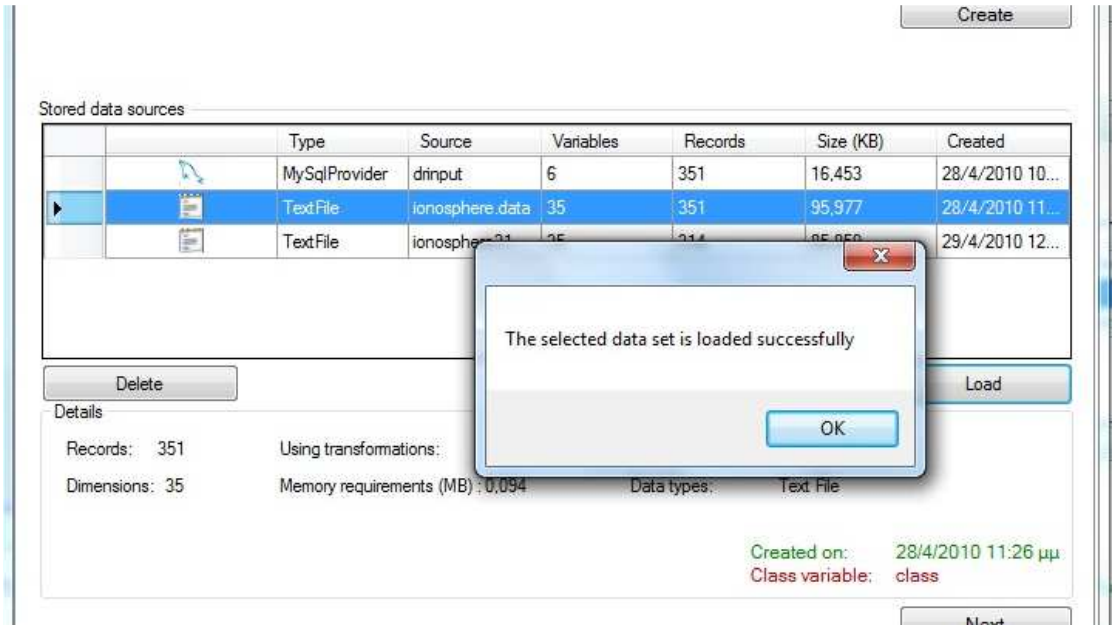


Figure 4.1.3.2 – Successful loading of stored dataset

with the configuration you have made on previous execution of the program

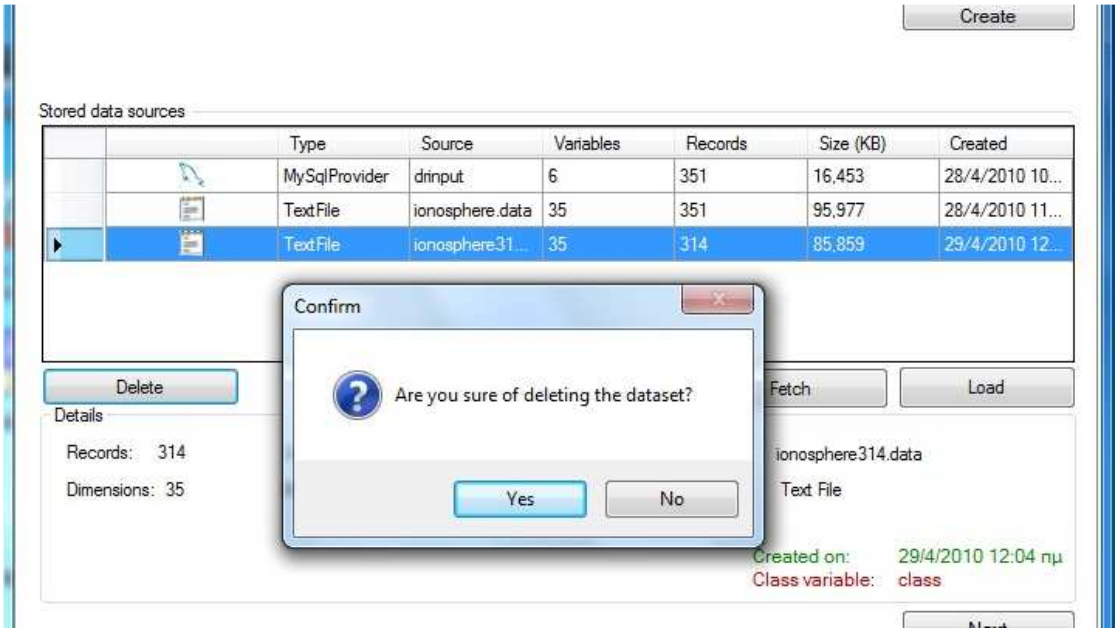


Figure 4.1.3.3 – Delete stored dataset

4.2. Second Level – Data Transformation

By pressing the "Next" button, one can proceed in the data transformation stage. The result screen appears in figure 4.2.1.

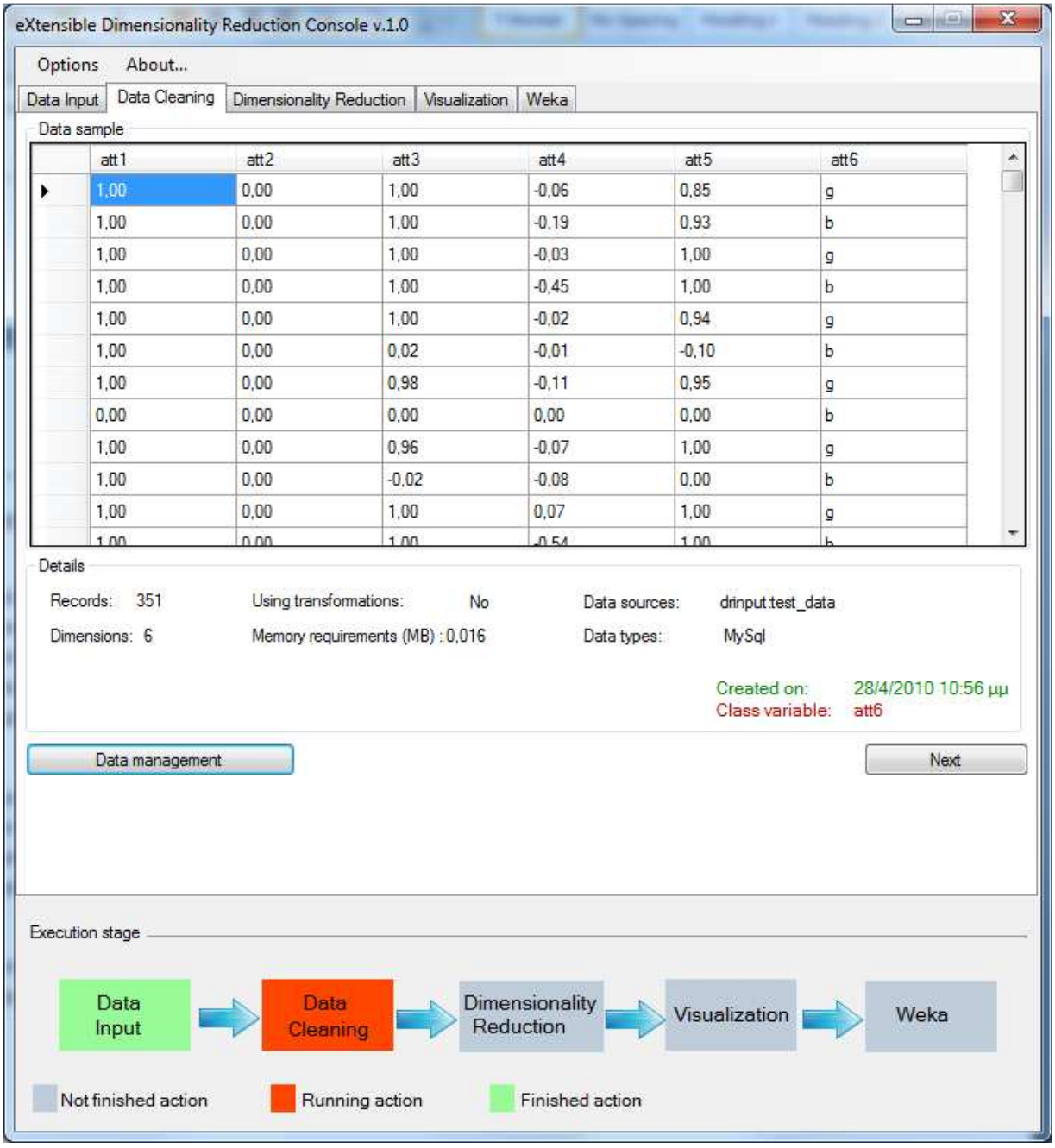


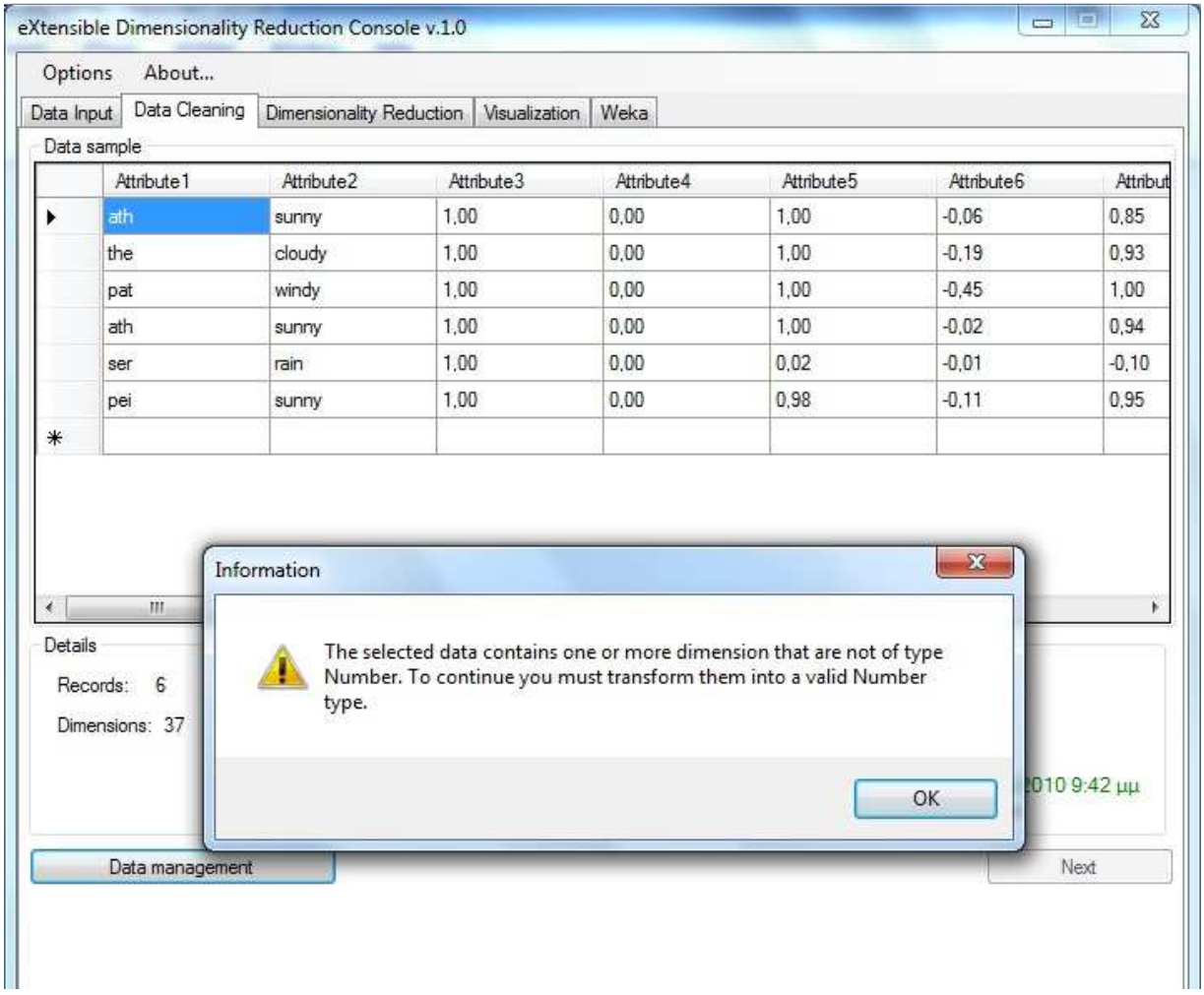
Figure 4.2.1 – Data management screen

The central frame of the screen shows a data sample extracted from the selected database. Right under the main frame, you can see the database details. The data that appears here is exactly the

same with the equivalent frame of the data input stage (see Pargraph 4.1 – fifth screen area). Under this frame you can see two buttons with the available options at this stage of application: “Data processing” or “Next”.

If you press the “Next” button, the application proceeds to the next level without any transformation or any change to the source data. This way you pass to the dimensionality reduction stage with all source data, without any modification. This is, of course, possible only if all data, except for the data class, is numerical.

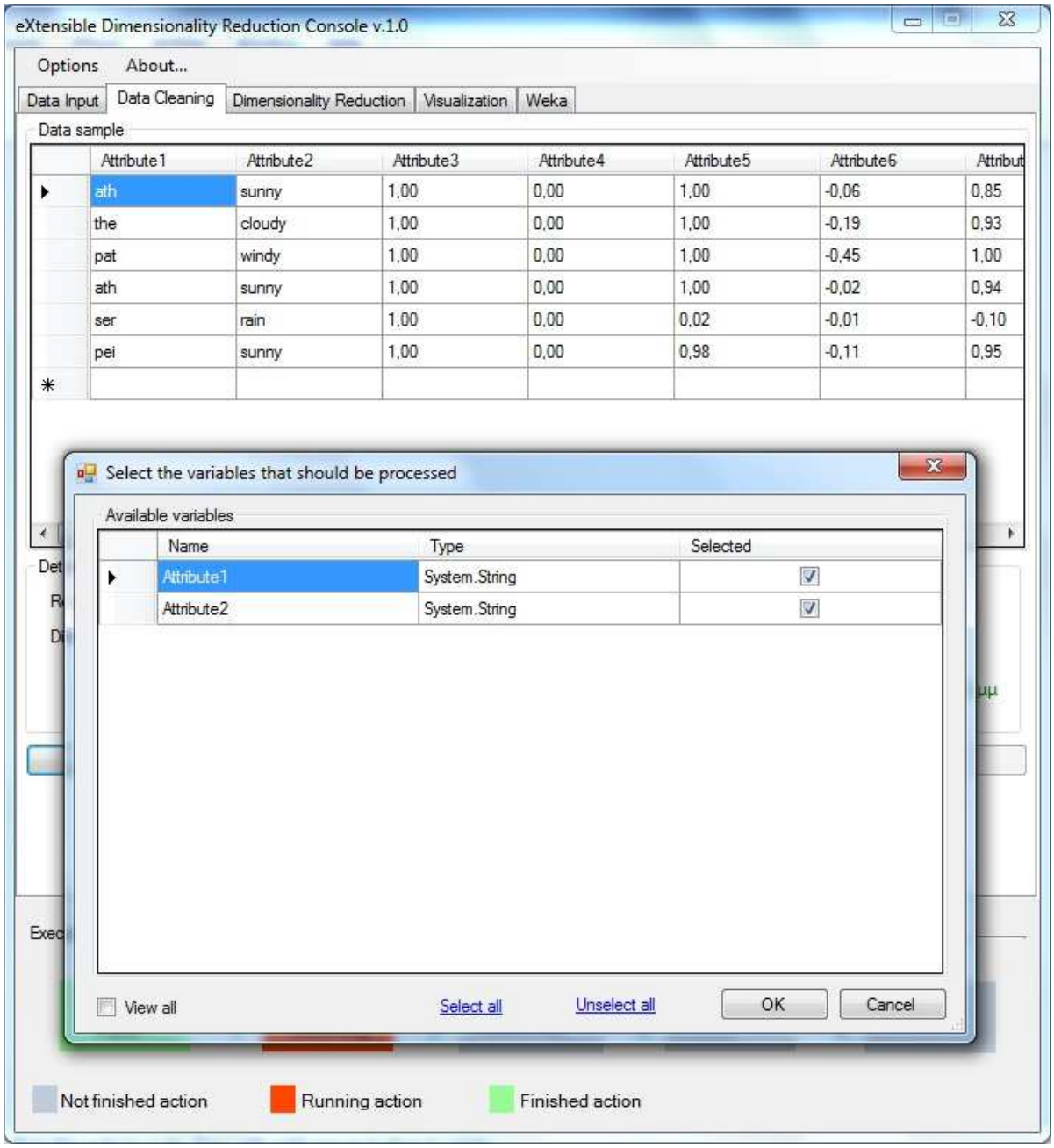
If you have chosen non-numerical data, the corresponding warning message will appear as follows (picture 4.2.2), informing you that you have to modify the data before the dimensionality reduction.



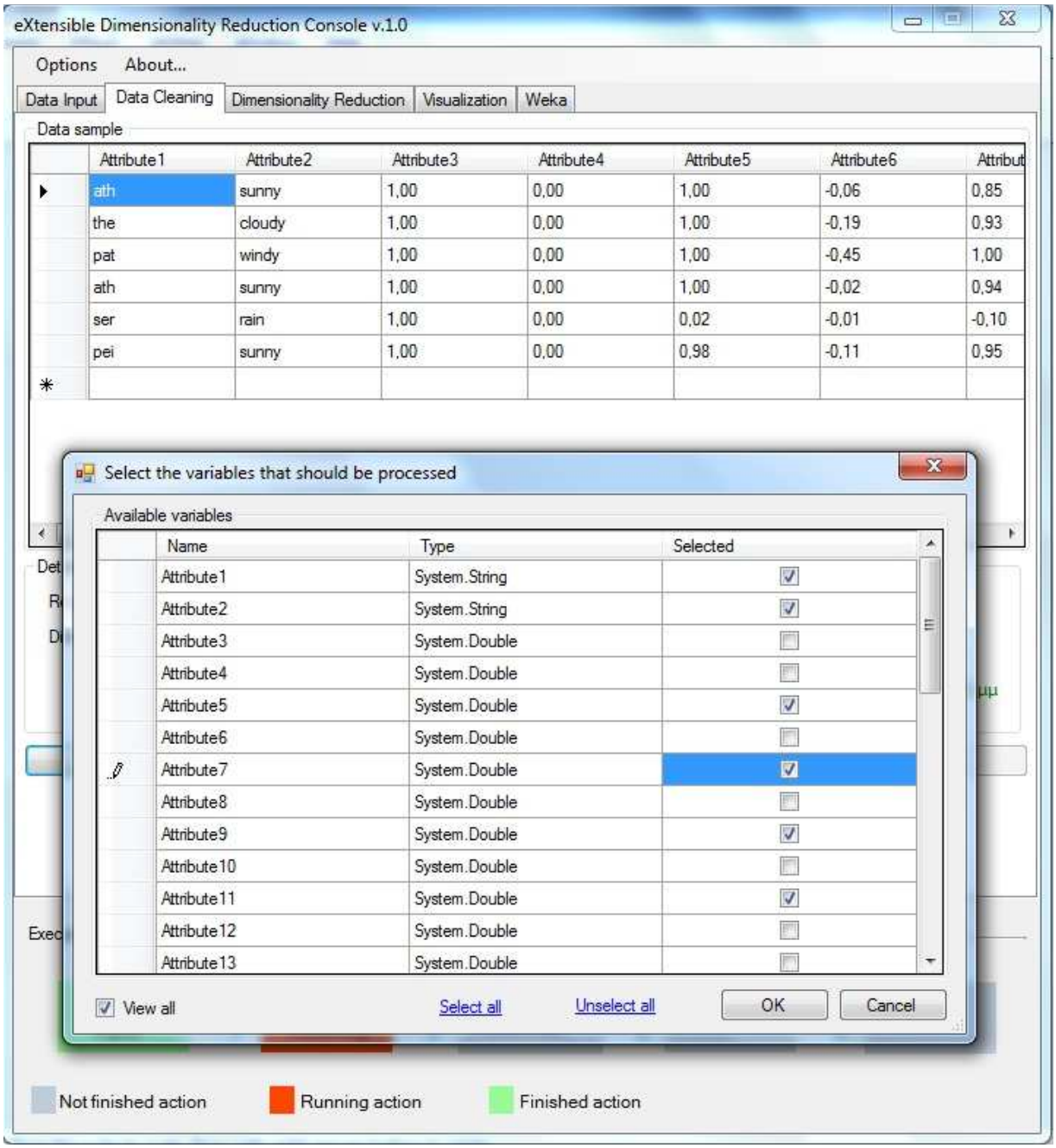
Picture 4.2.2 – warning for non-numerical data transformation

If you press the data processing button, the corresponding screen will appear. If there is non-numerical data for transformation, it can be pre-viewed on the screen (picure 4.2.3). Otherwise, the list is empty. By pressing on the lower left part the “View all” control button, you can see all database

dimensions (picture 4.2.4). You select those you want to transform and then press the “OK” button. Alternatively, you can press the “Cancel” button and go back to the central data transformation screen (picture 4.2.1) without having made any data change.

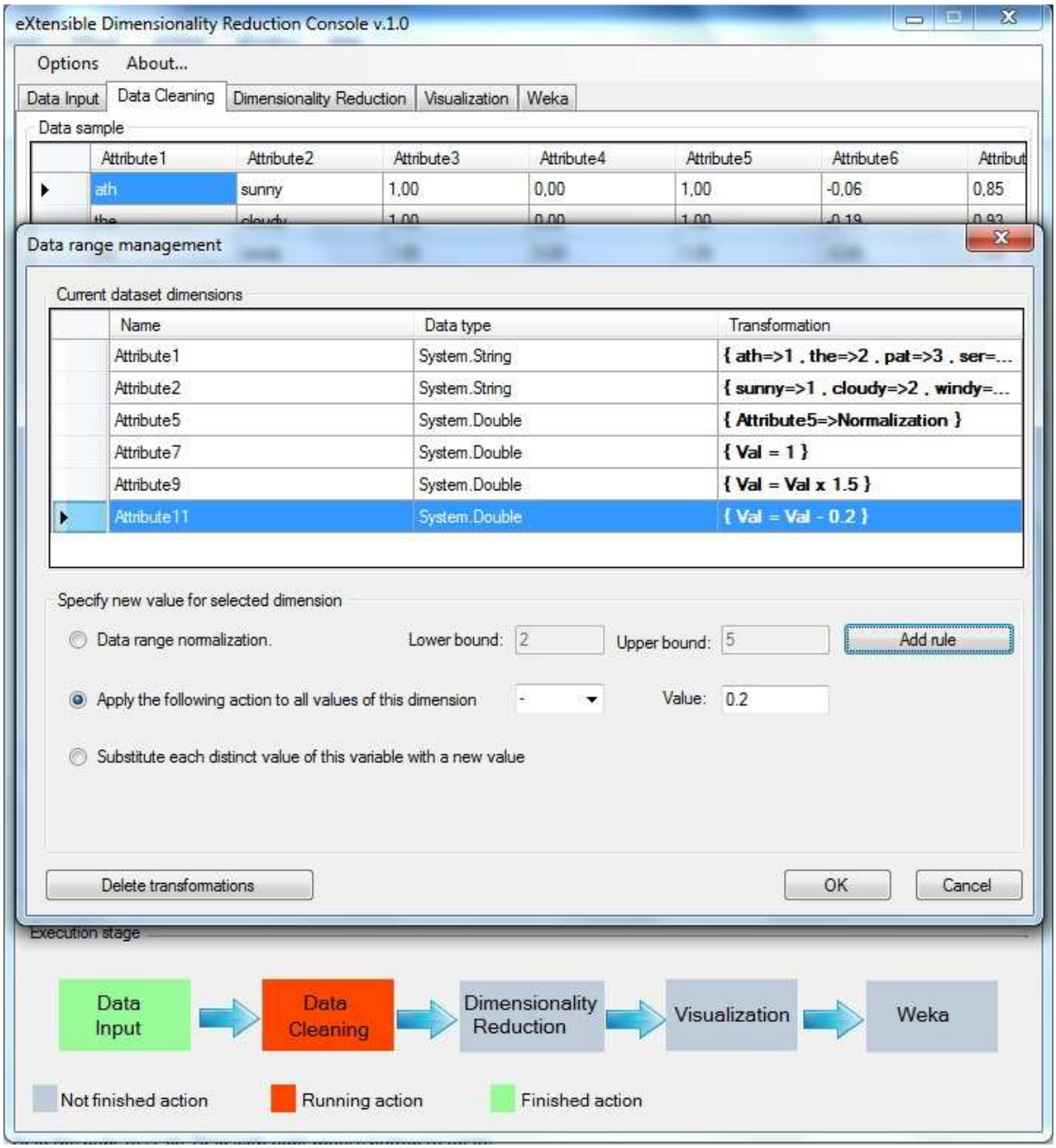


Picture 4.2.3 – Data selected by default for transformation



Picture 4.2.4 – Appearance of all data and selection of the ones to be transformed

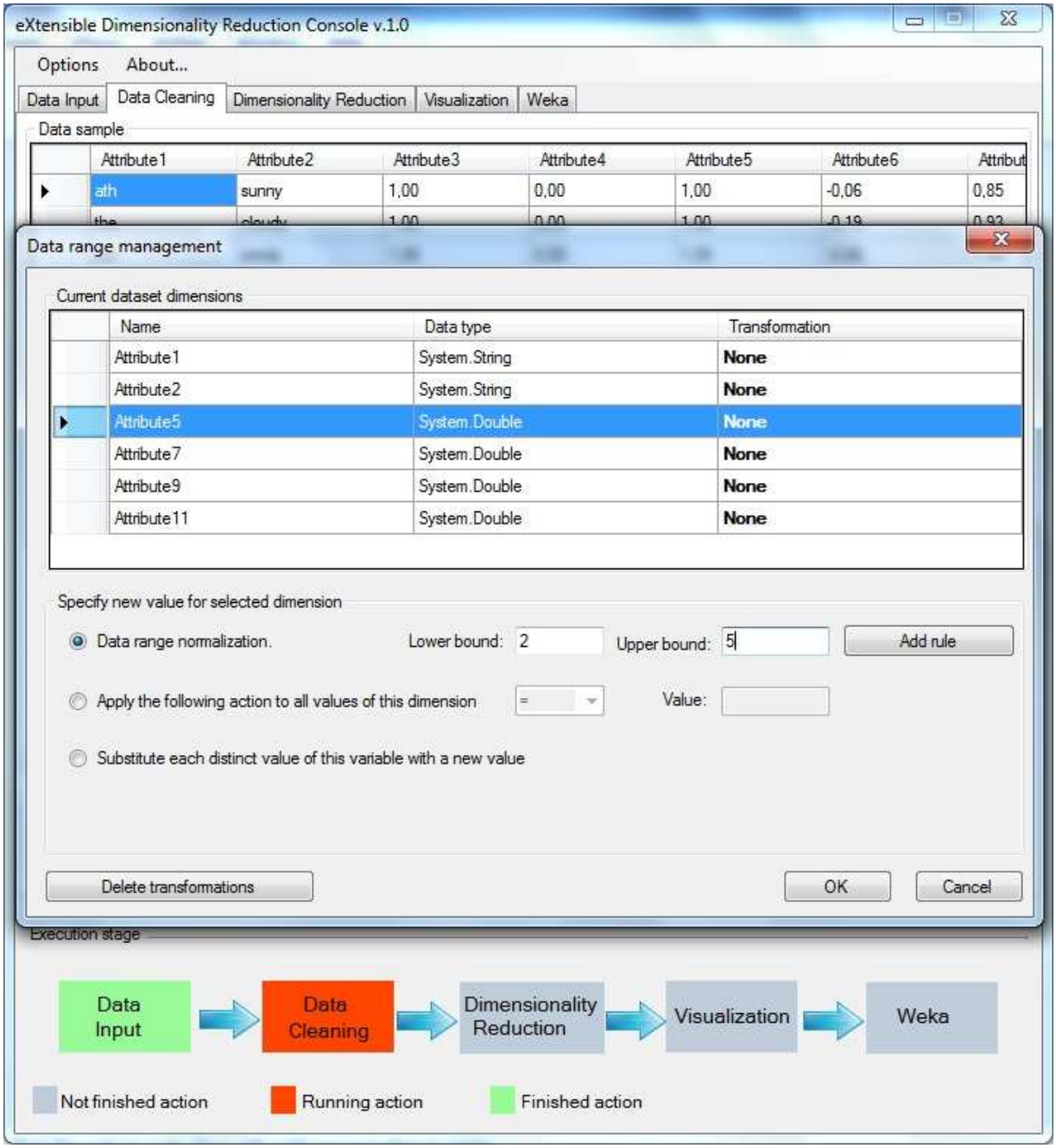
After having selected the columns you want to transform, the transformation window appears (picture 4.2.5), showing two major areas. The first one is the data list and the second one the available transformations' area. This list shows the name of the selected column, the data type, as well as the transformation rule you have set. Initially the transformations will be blank for all columns. In order for the transformations to be accepted, you have to set a rule for all selected variables (in the picture you have already selected all the transformations).



Picture 4.2.5 – Data transformation screen

There are three kinds of transformations that can be applied to data. Normalization (picture 4.2.6), mathematical operation (picture 4.2.7) and value substitution (picture 4.2.8). The first two can only be applied to numerical data, whereas the third one to characters as well. The value substitution is the transformation that has to be applied to all the columns that contain alphanumerical data.

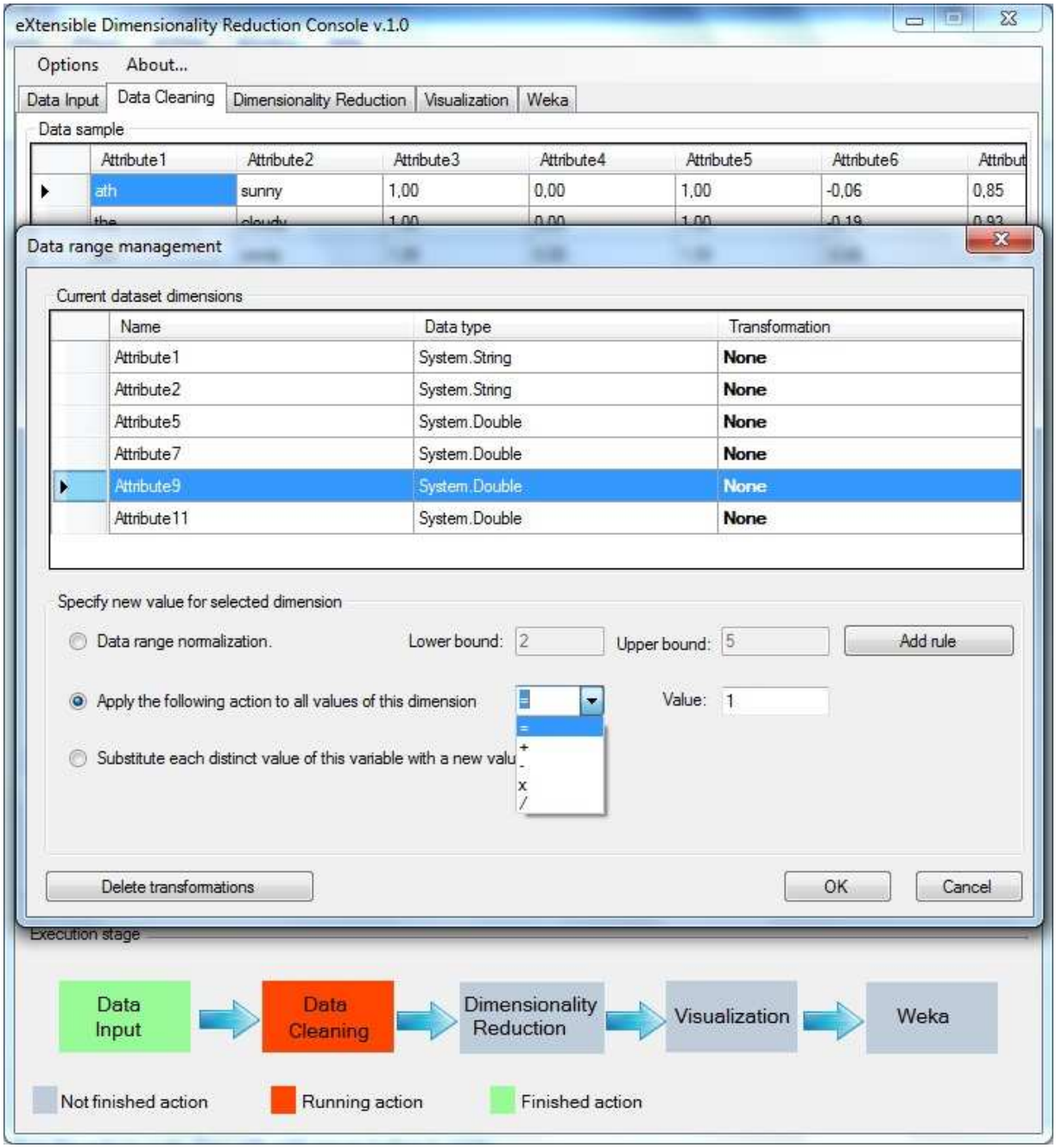
Normalization is the mapping of the selected valued onto a specifically set width. If normalization is chosen for non-numerical data, the application of the rule will be rejected and an error message will appear.



Picture 4.2.6 – Values normalization

The second transformation, which can also be applied only on numerical data, is the application of a mathematical operation on all selected values. It functions as follows: first, you select one of the

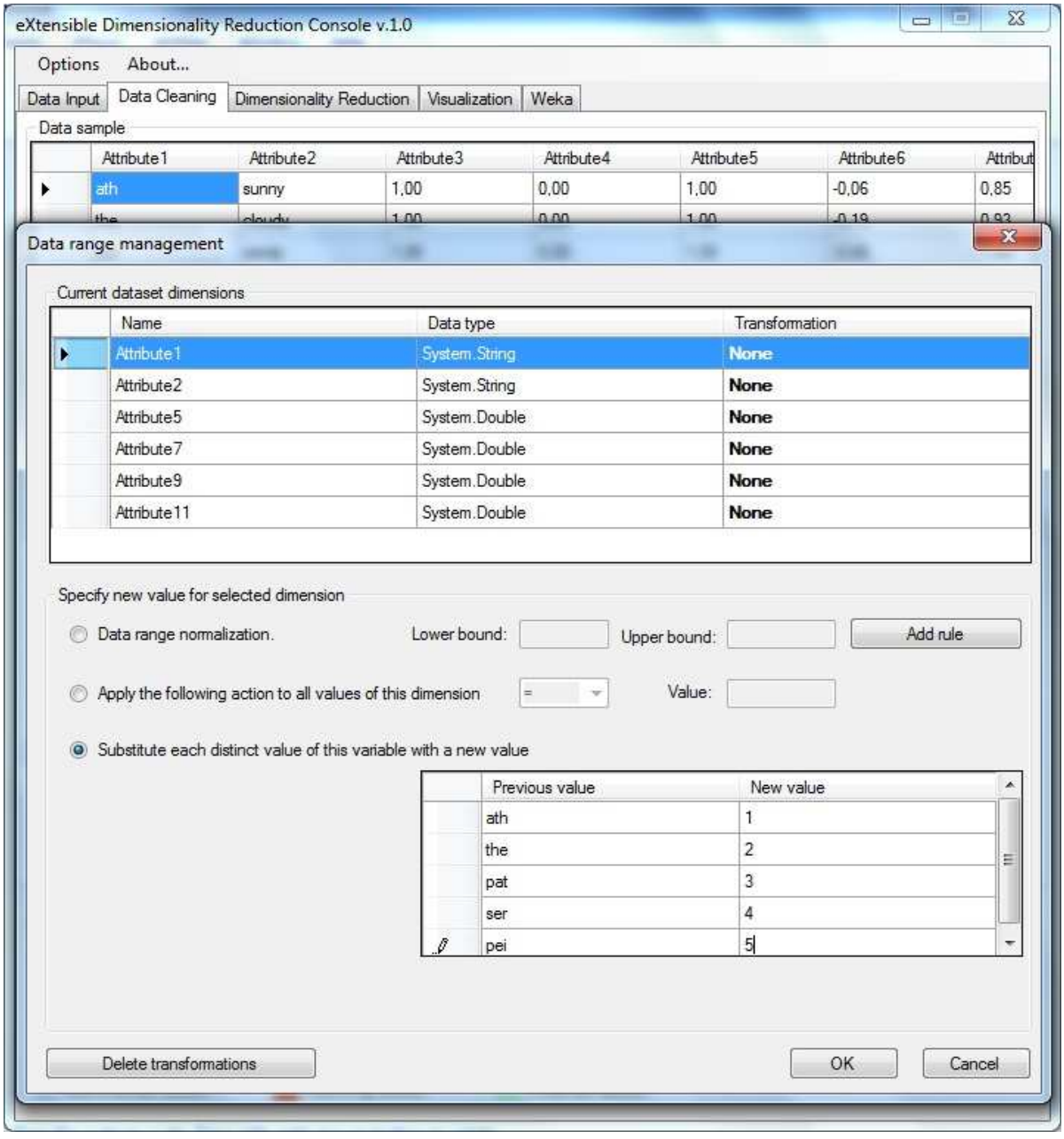
common mathematical operations (addition, subtraction, multiplication, division) and then you select a numerical value. Our rule will apply on the action that you set for all values.



Picture 4.2.7 – Action application to all dimension values

The last transformation is the substitution, the most common alphanumeric data filtering method. This way users can create an one-by-one correspondence between alphanumerical and numerical values. This action is very common in the data mining sector. When the user selects a variable from the list and presses on the distinct value substitution, a list with all distinct values of the dimension appears.

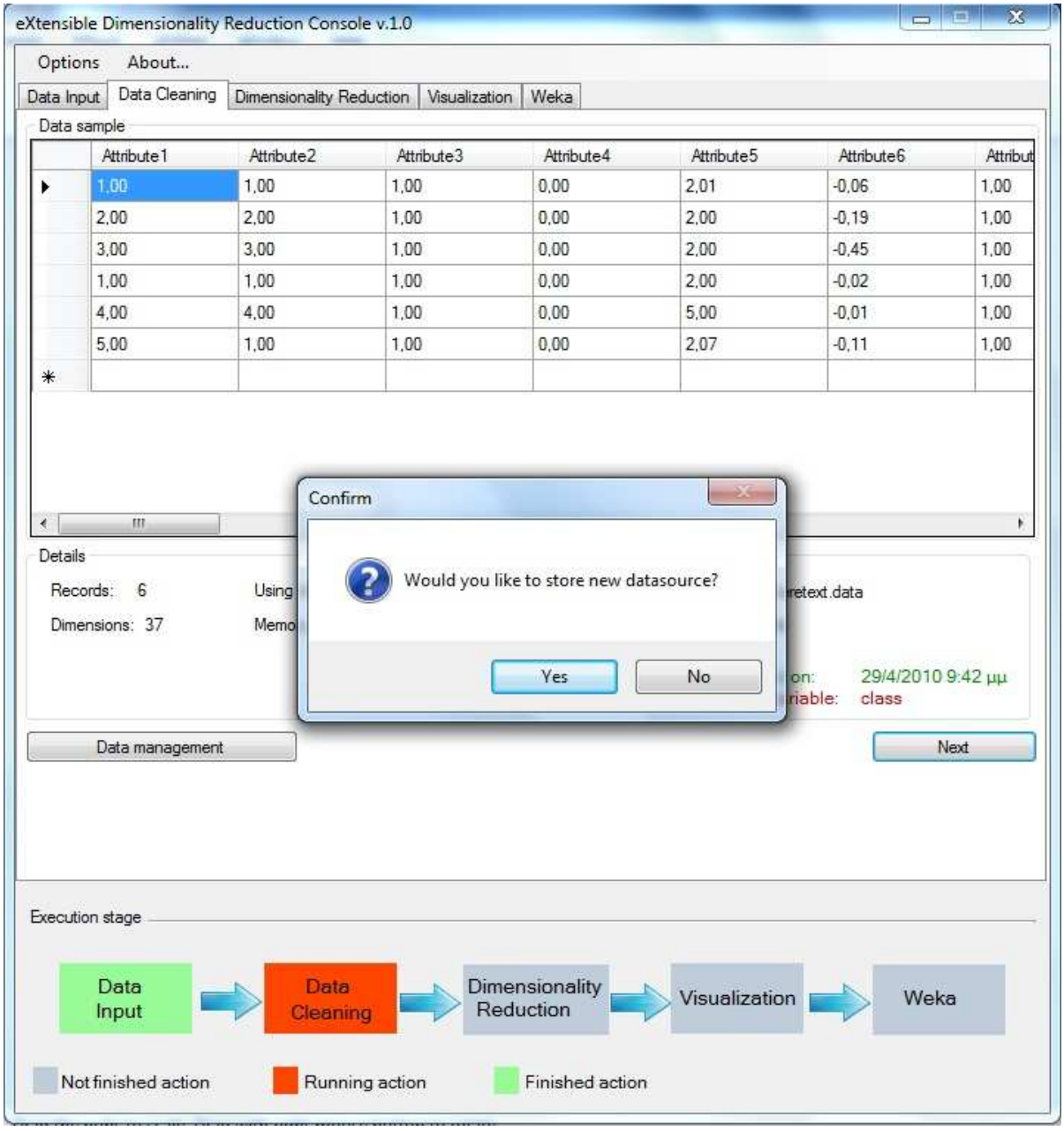
In this list users can set a new value for each of the current values. The new value can be only numerical. If an alphanumerical value is set, the application will show an error message.



Picture 4.2.8 – Values' substitution

As soon as the process is completed, all transformations that are going to apply to the dataset can be viewed on the upper-list of the screen. Thus, by pressing the "Delete transformations" button, users can delete all the transformations; by pressing "OK", they can apply the transformations on the data, whereas by pressing "Cancel", they can cancel the procedure and return to the previous screen. After executing the transformations, users are given the option to store the total dataset in the current

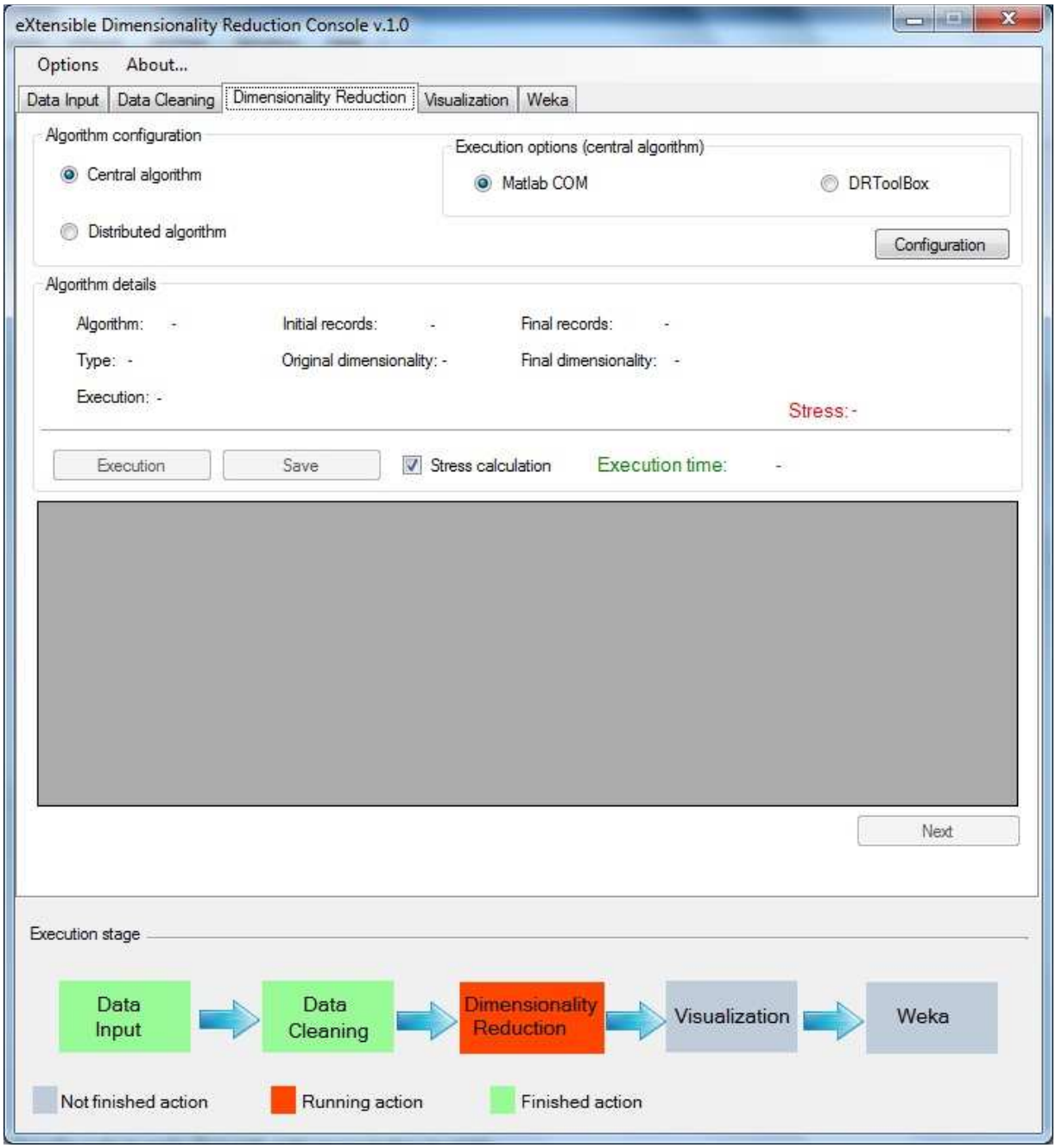
format (selected dimensions and transformations).



Picture 4.2.9 - Dataset storage

4.3. Third level – Dimensionality Reduction Algorithms Execution

This stage is the core of the program, since it describes the procedure for which the xSDR tool was created. As soon as the user has selected the data on which the process of dimensionality reduction will be applied and proceeds to the data management stage, he will see the following application screen:



Picture 4.3.1 – Central dimensionality reduction screen

Initially, the user has to choose the algorithm type that is going to be executed. The options are central and distributed. The configuration and execution are described in paragraph 4.3.3 of the guide. If the user selects a central algorithm, he will have the following additional options: he can execute an algorithm from DRToolbox or a custom algorithm (written in Matlab language).

DRToolbox is a collection of the most well-known dimensionality reduction algorithms. The user can choose every available algorithm, modify its set-up and execute it, as it is analytically described in

paragraph 4.3.1. The link for toolbox's website is available and can be found in the guide's annex. For this exemplar, the latest version of DRToolbox was used (up to the day of the paper's completion); however, taking into consideration the regular update of algorithms, we encourage a frequent control of the website. For a full description of the integration, see paragraph 4.3.1.1 – "New DRToolbox versions integration".

Custom algorithms execution is, as mentioned before, an additional option of the program. Along with the program installation process, a number of algorithms is also pre-installed. Their configuration and execution is described thoroughly in paragraph 4.3.2. In addition, the import of new algorithms is possible. Algorithms should have a certain form and to follow the rules set in paragraph 4.3.2.1., "Custom algorithm writing".

As soon as the user selects and sets up the algorithm he is going to apply, the "Execution" option is activated and the dimensionality reduction can now take place. Simultaneously, there is the possibility to calculate the stress between the initial and the reduced data sum. With the procedure completion, the user will be informed, by means of a "Message Box", on the successful, or not, application of the algorithm. After a successful application, the results storage option is activated. The storage can be made in .arff or simple .txt format.

4.3.1. DR Toolbox Algorithm Execution

The execution algorithm of DR Toolbox is as follows: choose from the dimensionality reduction step main screen (figure 4.3.1) "Central Algorithm", select the "DRToolBox" and then press the button "Configuration", that shows the available algorithms of DRToolbox, as presented in Figure 4.3.1.1.

From the left column of the screen you can choose the algorithm that you want to execute and on the right part of the screen the available parameters will be displayed. At this point, you should refer to the parameter "Eigenanalysis Implementation", which is related to the way the library performs an algorithm, is found only in DRToolbox algorithms and is available for all algorithms. This parameter lies in the Matlab option, if the selected dataset contains fewer than 10,000 entries; in case the selected dataset contains more than 10000 entries, click JDQR.

By pressing the "OK", the algorithm is parameterized and ready for execution (Figure 4.3.2.2). The algorithm execution process and the messages shown are the same for all types of algorithms and will be described in paragraph "4.3.2 - Custom Algorithms Execution"

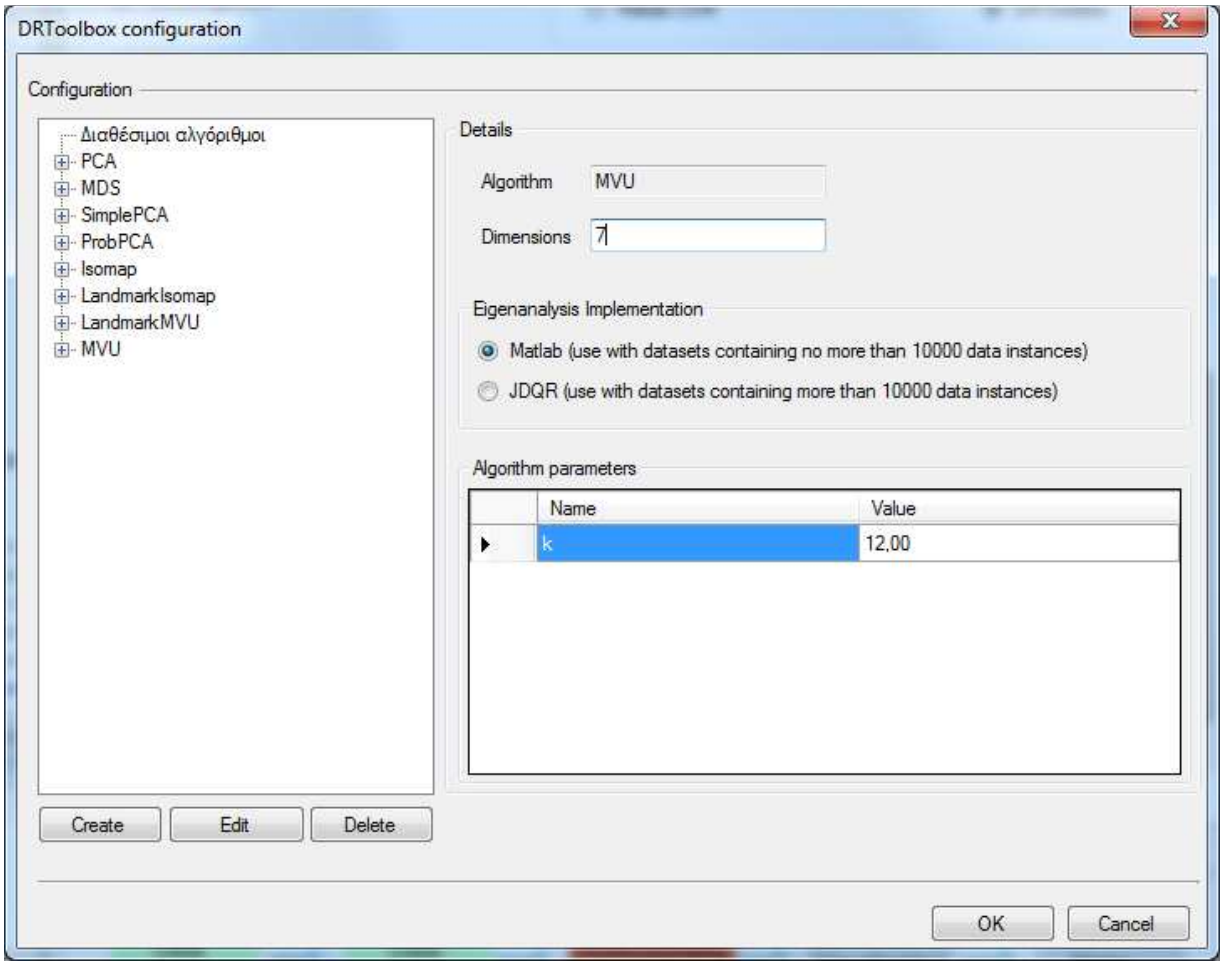


Figure 4.3.1.1 –DRToolbox algorithm configuration

4.3.1.1 Incorporation of new versions of DRToolbox

One of the measures proposed to increase the use of the tool is to attach the program to the new versions of DRToolbox. The procedure to be carried out by the user is the following: through the deployment tool of Matlab, he/she must build a dll library file and insert it into the program. The necessary steps are described in details below.

After rebooting, open the Matlab command line program. There, type the command "deploytool". This command will open the GUI Deployment tool of Matlab. Then select "New project". We recommend that you give your project and your class names, as suggested below, in order to perform changes only in the Reference File and not in the source code as well.

Figure 4.3.1.1 shows the initial screen of the Deployment Tool in Matlab. We choose to first create a new.Net Assembly and call the project DRToolboxWrapper; then, we create a new class called

DRToolboxWrapperclass and add it to the Class compute_mapping.m file from folder DRToolBox you have already downloaded. This file contains all the information needed, regardless of the algorithms that we're going to use. Therefore, you do not need to add another file to the class.

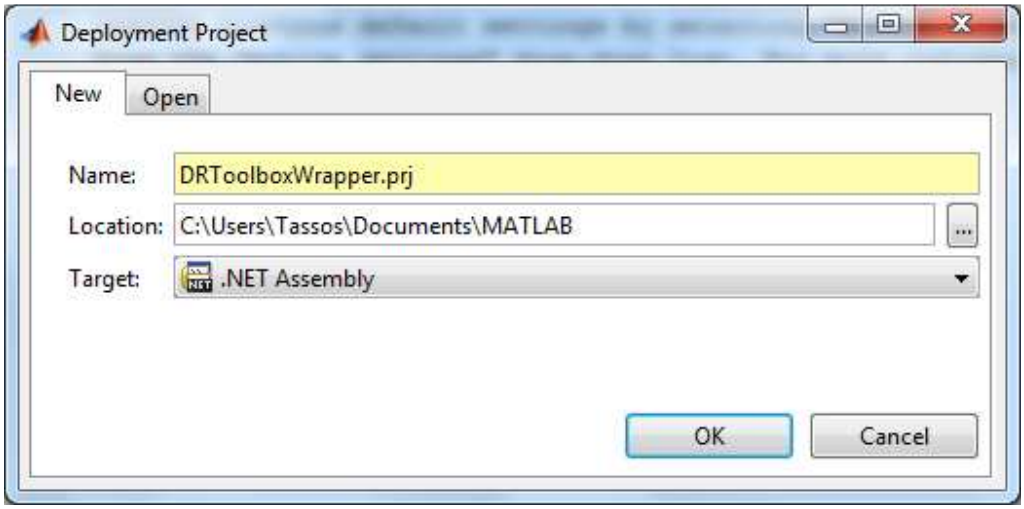


Figure 4.3.1.1 – New project on deployment tool for Matlab

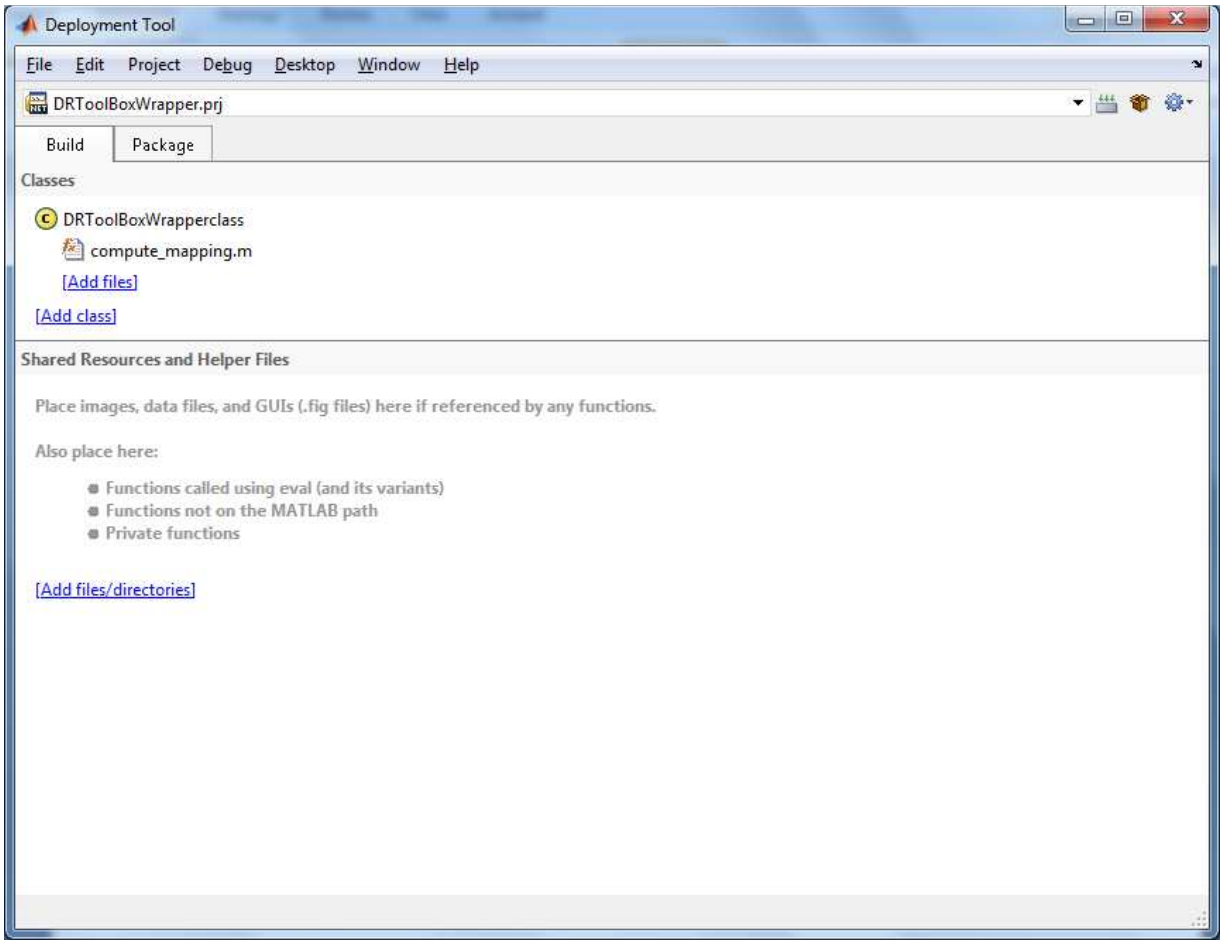


Figure 4.3.1.2 – Project Details

Tab "Package", allowing you to see the contents of the package (Figure 4.3.1.3). The necessary files are. Dll and .Ctf file (component technology file). These are to be included in xSDR application, to achieve the desired functionality. In addition, you must include the MCR Installer in our package, if this is a new DRToolbox assembly. As already mentioned, there is the need for a Matlab Compiler Runtime version, created from the same system that the .dll file is created, in order to function properly. It goes without saying that all MCR produced by the same system do not vary, so if you have already created the MCR earlier, then you do not need to include it in the package.

After setup and configuration of the package contents, you can proceed to compile and create the assembly file. Then, select "Debug" -> "Build" from the menu on the Deployment Tool and expect to complete the process. The application screen that you are to see, should look like the one shown in figure 4.3.1.3.

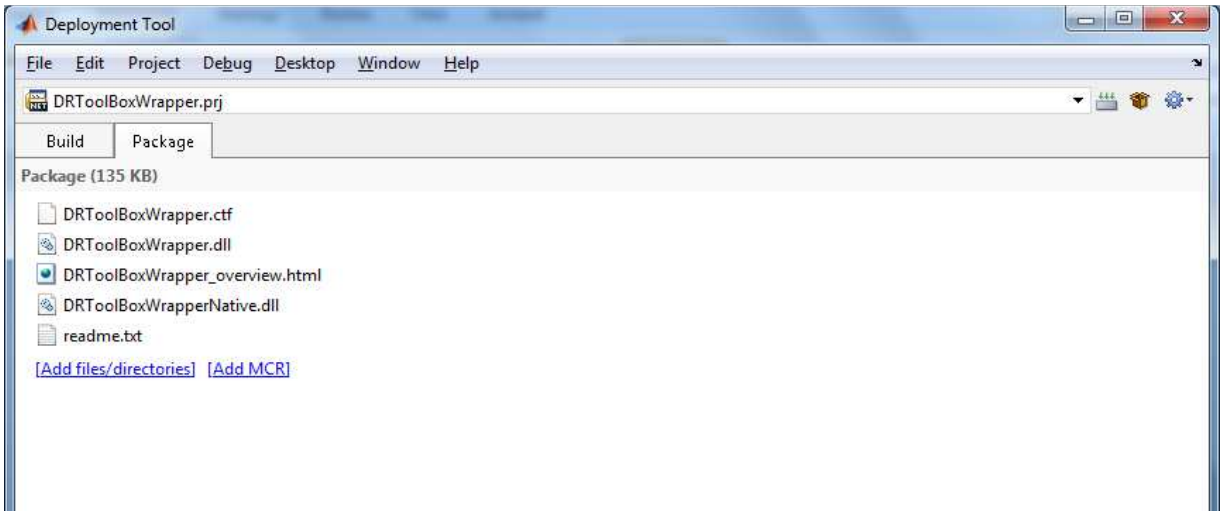


Figure 4.3.1.3 – Package Details

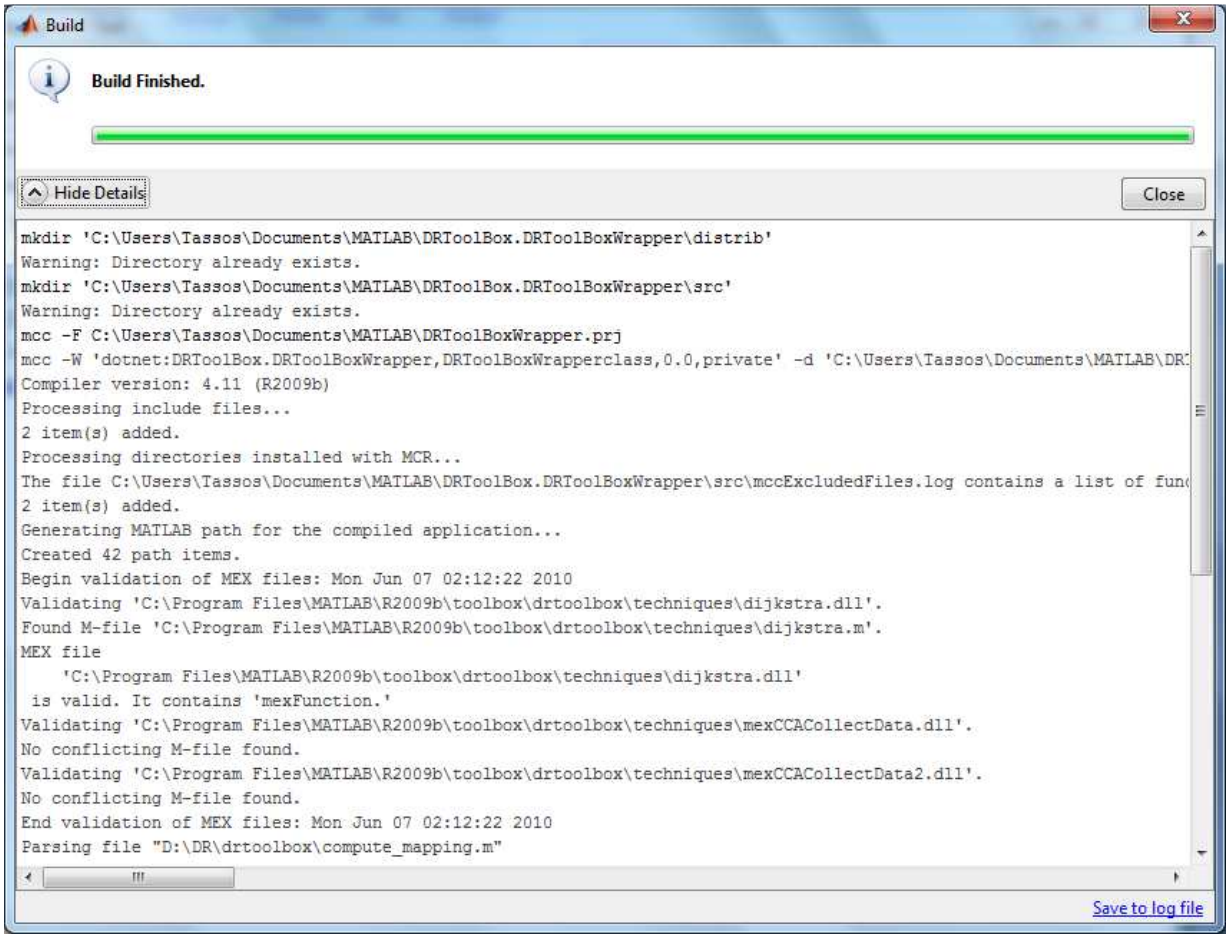


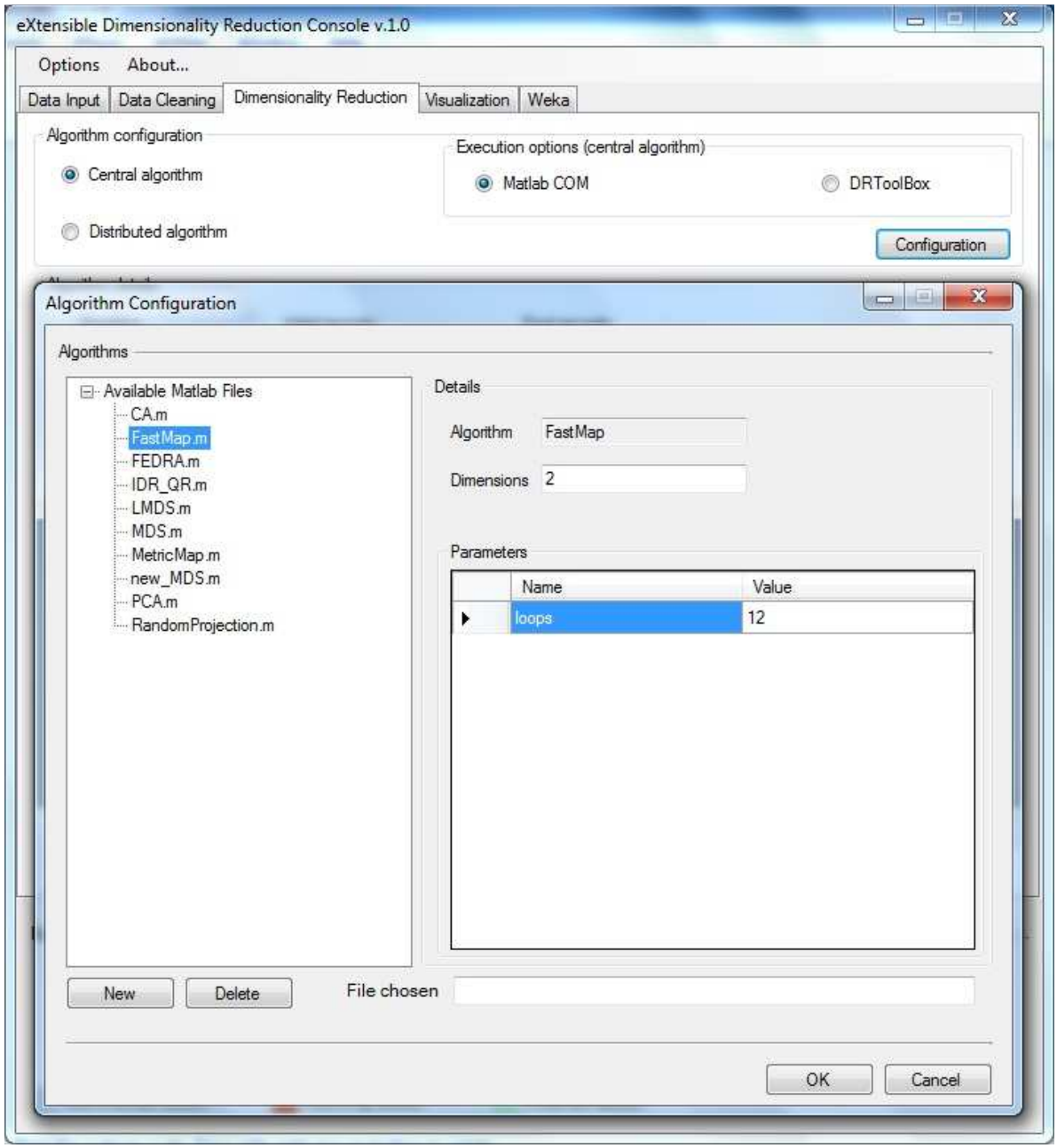
Figure 4.3.1.4 – Successful Assembly building using the Deployment Tool

Naturally, there is the option to build the assembly without using the deployment tool, just by executing the right commands through the matlab command line. Below you can see the commands used for our assembly.

```
mkdir 'C:\Users\Tassos\Documents\MATLAB\DRToolBox.DRToolBoxWrapper\distrib'

mkdir 'C:\Users\Tassos\Documents\MATLAB\DRToolBox.DRToolBoxWrapper\src'

mcc -F C:\Users\Tassos\Documents\MATLAB\DRToolBoxWrapper.prj

mcc -W 'dotnet:DRToolBox.DRToolBoxWrapper,DRToolBoxWrapperclass,0.0,private'
-d 'C:\Users\Tassos\Documents\MATLAB\DRToolBox.DRToolBoxWrapper\src' -T
'link:lib' -v -C
'class{DRToolBoxWrapperclass:D:\DR\drtoolbox\compute_mapping.m}'
```

4.3.2 Custom algorithms execution

The execution of algorithms stored in Matlab files is made as follows: select a central algorithm through the dimensionality reduction stage screen, followed by the selection of the Matlab COM use. then, press on “Configuration” button and the corresponding screen appears (figure 4.3.2.1).



Picture 4.3.2.1 –Custom algorithms configuration

On the left part of the form, you can see a list of all available algorithms. By choosing one of them, you should be able to see on the right part of the screen the name of the selected algorithm, the sample dimensions after the reduction and finally the available parameters. On the lower part of the screen, you can locate the “New” and “Delete” buttons. By pressing on each, you can import a new algorithm or delete an existent one.

After the configuration, the algorithm is ready to be executed. By clicking "OK", go back to the original screen of dimensionality reduction, where you can now see more details of the algorithm to be executed, as previously defined by the configuration screen (Figure 4.3.2.2). Pressing "Execution" starts the execution of the algorithm. The performance depends on the system running and the size of the dataset and can last a few seconds. After the process is completed successfully, the new reduced dataset will appear and the application will display the corresponding message of successful algorithm execution (Figure 4.3.2.3).

After a successful execution of a dimensionality reduction algorithm, the user is able to store the new data (Figure 4.3.2.4). Storage can be done using .Arff files or simple text files. Arff format is ASCII text files that were created solely for use by the data mining application Weka. More on this type of file can be found at http://www.cs.waikato.ac.nz/~ml/weka /arff.html. By selecting Save in text format, the user is given the option to choose the separator between columns (Comma, Space, Tab or any other user defined separator). After saving, if the option «Store file and open it» is selected, the file will be opened for reading by default application (Weka & Notepad respectively).

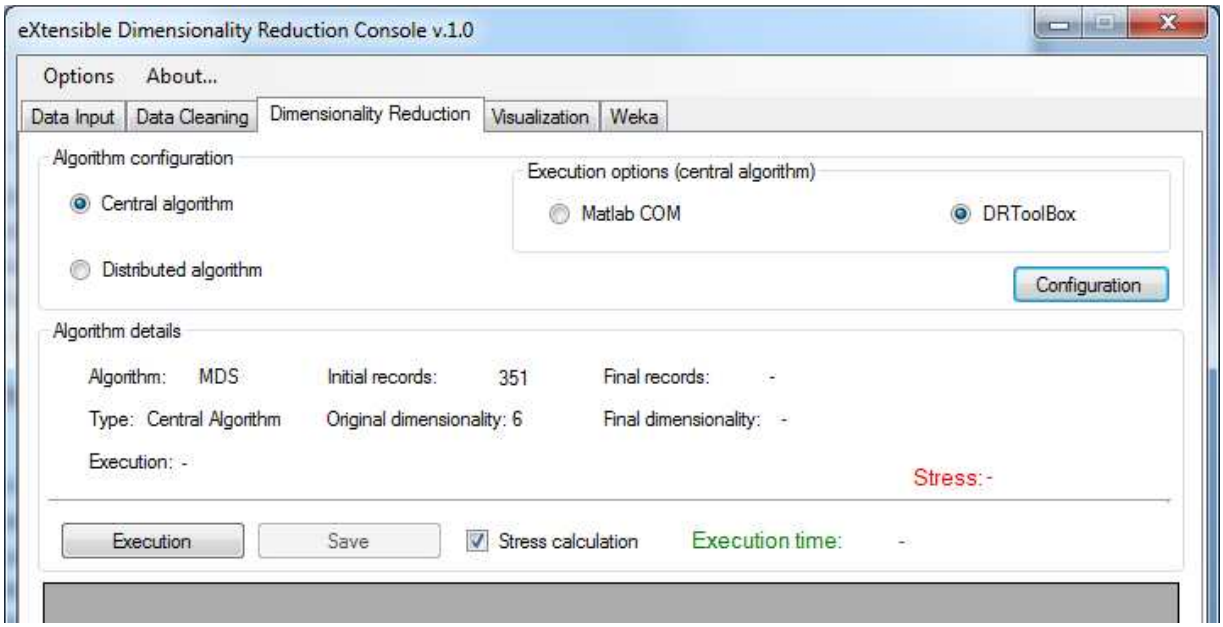


Figure 4.3.2.2 – Configured algorithm ready to be executed

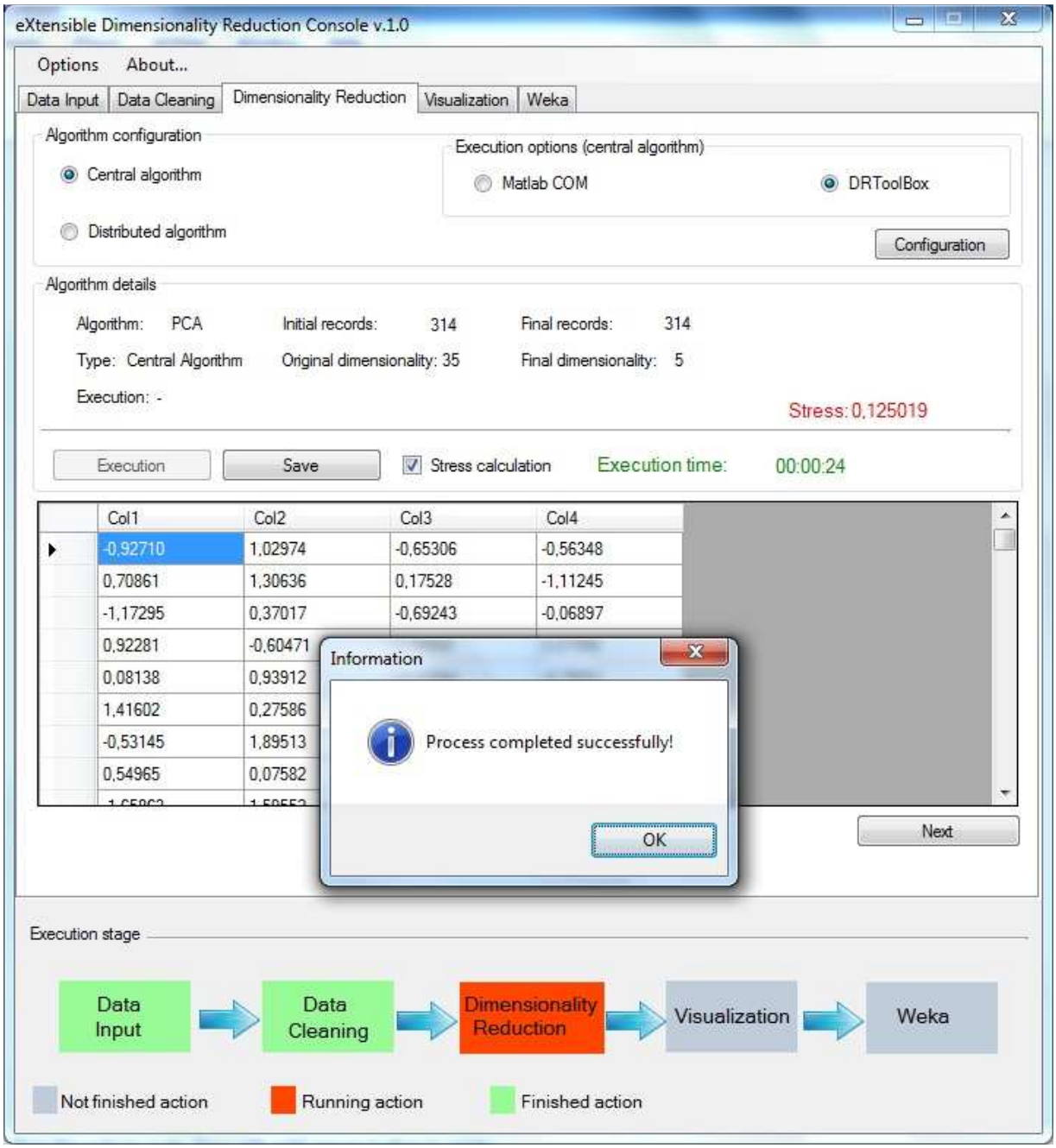
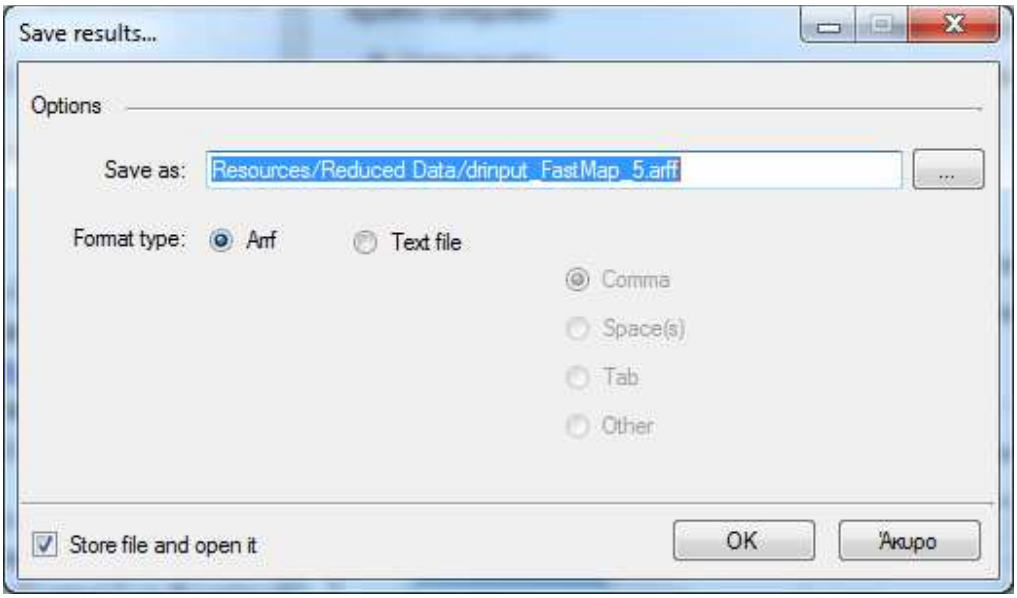


Figure 4.3.2.3 – Successful execution of dimensionality reduction algorithm



Εικόνα 4.3.2.4 – Αποθήκευση νέου συνόλου δεδομένων

4.3.2.1. Custom Algorithm writing

As mentioned before, xSDR gives users the possibility to import their own dimensionality reduction algorithms. Below we will present the pattern, according to which an algorithm has to be written to be properly executed by the xSDR application. The algorithm must be written in Matlab language. On the upper part of the text, there must be a XML header, which contains information that the program will use for the execution.

The algorithm should accept as input a two-dimensional array - which is the data field to perform the dimensionality reduction algorithm, the number of dimensions of the final range and the location of the class data in the range. The algorithm gives a two-dimensional table as output, which is the final data set after the dimensionality reduction. All the additional parameters of the algorithm must be given within the header.

Details for the header: This header contains information to be used by the application. All information entered in the header should be commented on, so as to be ignored by the Matlab Compiler and not displayed as “error” during the execution. The format of the header, along with the body of the algorithm, are shown below. The name and description are prerequisites and must in no case be omitted. Then the user can set the parameters he wants. These parameters must comply with the corresponding form of the program. Each parameter must contain three fields: name type and finally value.

Below we present a sample code, that will be implemented by each algorithm, and then, in figure 4.3.2.1.1., some sample codes from the implementation of the algorithm Metric Map, as projected through the Editor of Matlab.

```
%<algorithm>
%<name>AlgoName</name>
%<description>Implementation of <AlgoName>algorithm.</description>
%<parameters>
%<parameter>
%<name>ParameterName</name>
%<type>ParameterType</type>
%<value>DefaultValue</value>
%</parameter>
%</parameters>
%</algorithm>
function [A]=PCA(k, classVariable)
   load 'inputData.txt';
   A = inputData;
.
Implementation of the algorithm
.
end
```

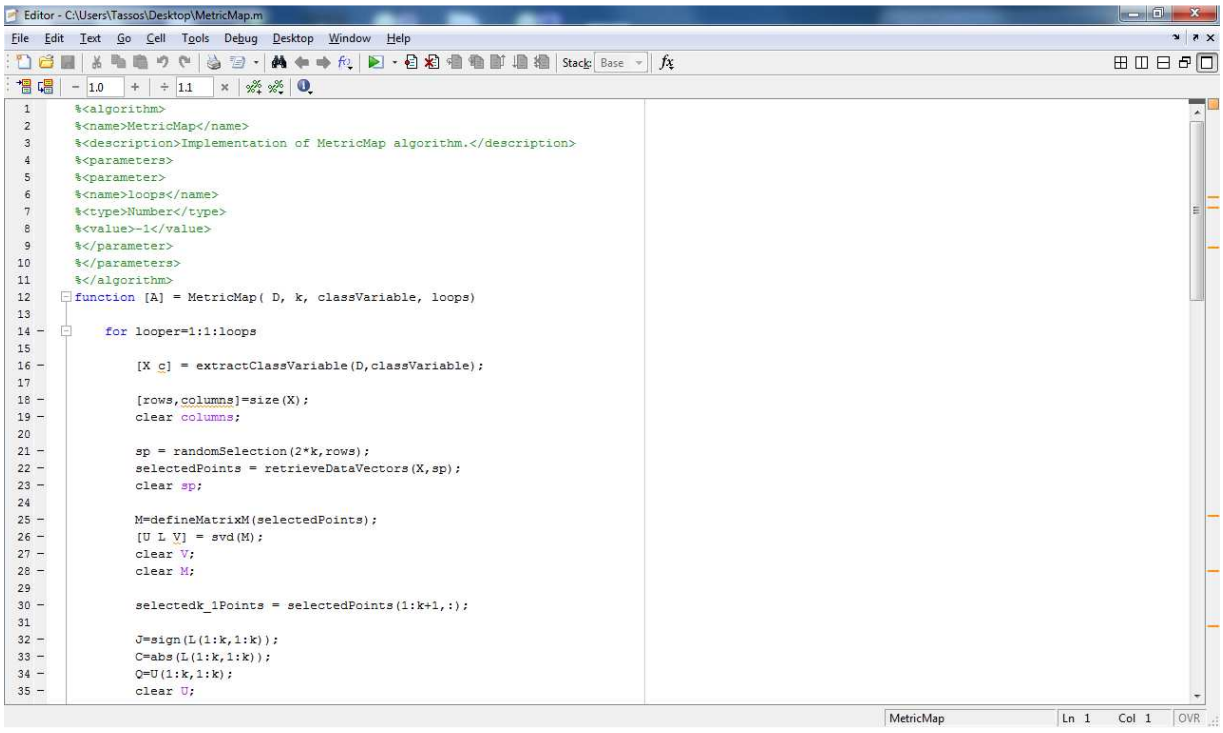


Figure 4.3.2.1.1 –Custom algorithm through the Matlab editor

4.3.3 Distributed Algorithms Execution

The execution of distributed algorithms does not differ significantly from the implementation of the key algorithms. When you move in the level dimensionality reduction (as shown in Figure 4.3.1), select “Distributed Algorithm” and press Configuration. In this way you move to the well known screen of algorithms configuration. From there, configure the algorithm as described in previous sections and do the execution, as described in Section 4.3.2

4.4. Fourth Level – Results Visualization

For results visualization, we have used Microsoft MSCharts library. This library has already been chosen in the initial creation of DRC, due to its completeness in diagrams and visualization means, its excellent cooperation with the other parts of the program (the library is exclusively designed for .Net Framework 3.5, which was used for the creation of the application) and, last but not least, its rapid creation of graphic graphs with minimum memory consumption.

In order to look the visual attribution of both the initial database and the database after the dimensionality reduction, pass to the next stage. The screen that appears on entering this stage is the one in figure 4.4.2. At this point, we have to mention that, in order for the database visualization to be executed, users must have set in a previous execution stage the "data class" of the dataset.

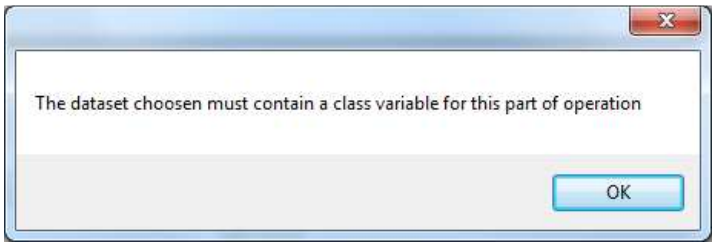
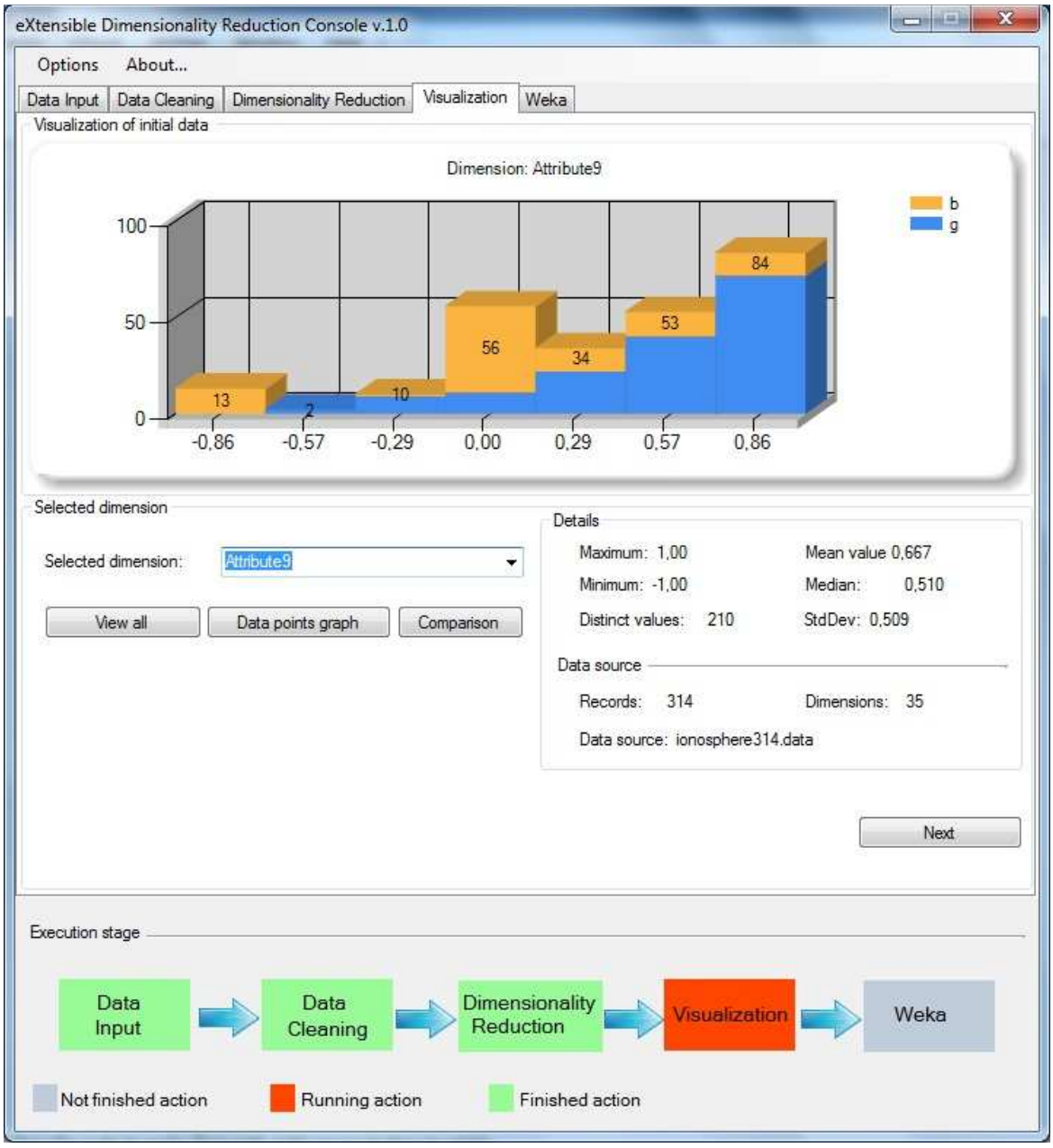


Figure 4.4.1 – The dataset does not contain data class and its visualization is not possible



Εικόνα 4.4.2 – Main screen of results visualization

On this screen you can see three major areas: in the first, central area of the screen, we find the graph of the initial database. At this point, the selected dimensions appear through the "drop-down" menu in the second area of the screen. Out of this selector, you can choose the dimension that you wish to project. The details regarding the dimensions appear it the third area of the screen. There you can see all the information related to the projected database. The information that is shown contains the following:

- Maximum value of selected dimension
- Minimum value of selected dimension
- The number of distinct values of the selected dimension
- The average value of the selected dimension
- The typical declination of the selected dimension (StdDev)
- The number of entries of the selected dimension
- The number of dimensions of the selected database
- The name of the selected database

Finally, as you can see under the dimension selector, there are three further viewing options. We will analyze these options in the forthcoming paragraphs.

4.4.1 "Select all" option

With the “select all” option, you have the column graphs for every dimension on the screen. This projection can be seen in figure 4.4.1. Through these graphs, you are able to compare the database variation after the dimensionality reduction. On the horizontal axis of the graph, you can find the values that belong to the selected dimension, whereas on the vertical one the appearance frequency of each value. The different colors correspond to the class that the points belong to. The legend situated in the upper right part of the screen explains which call is depicted by which color. Finally, each column contains the number of the value appearance.

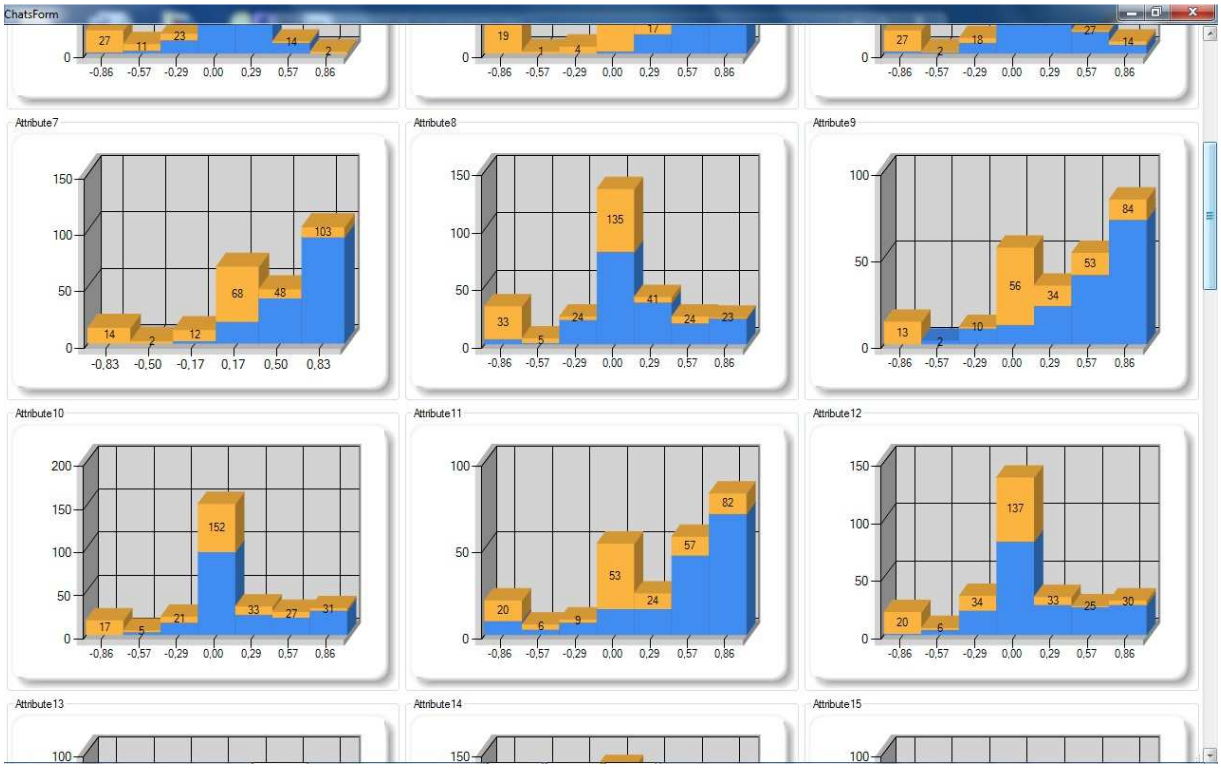


Figure 4.4.1.1. – “Select All” Option

4.4.2. Data Points Graph

By pressing on the "Data Points Graph", the data points graphs appear for all dimensions. Through this option you can explore the correlation between the two dimensions. This option will produce a graph table. The lines and the columns of the table are the selected dimensions. This way you can see how each dimension is correlated to the other. On the left part of the screen and under the graphs, you can find a selector. Through this menu, you can choose the dimensions you wish to project. On the right part of the screen there are the depiction settings. In particular, you are able to choose the size of the graph and the size of each point. Of course, for every change you have to press the "Apply" button, for the changes to take place (figure 4.4.2.1: graph table).

Then by double clicking on any data graph shown, you are able to see a magnified data points graph. As shown in figure 4.4.2.2, through this window you have the option to change the dimensions given and choose any other you wish. In addition, you can do "benchmarking". The view is displayed by pressing the "Visualize" button, located on the right side of the screen; by pressing the button, the screen will appear, as seen in Figure 4.4.2.3.

Through this window you can have a view of two different point graphs. On the left graph you have essentially the same with the previous projection. The difference at this point is that you can project on the adjacent axis system the dimensions resulted after the application of the dimensionality reduction algorithm. This fact renders this projection very useful, since it allows a clearer view of the transformation procedure.

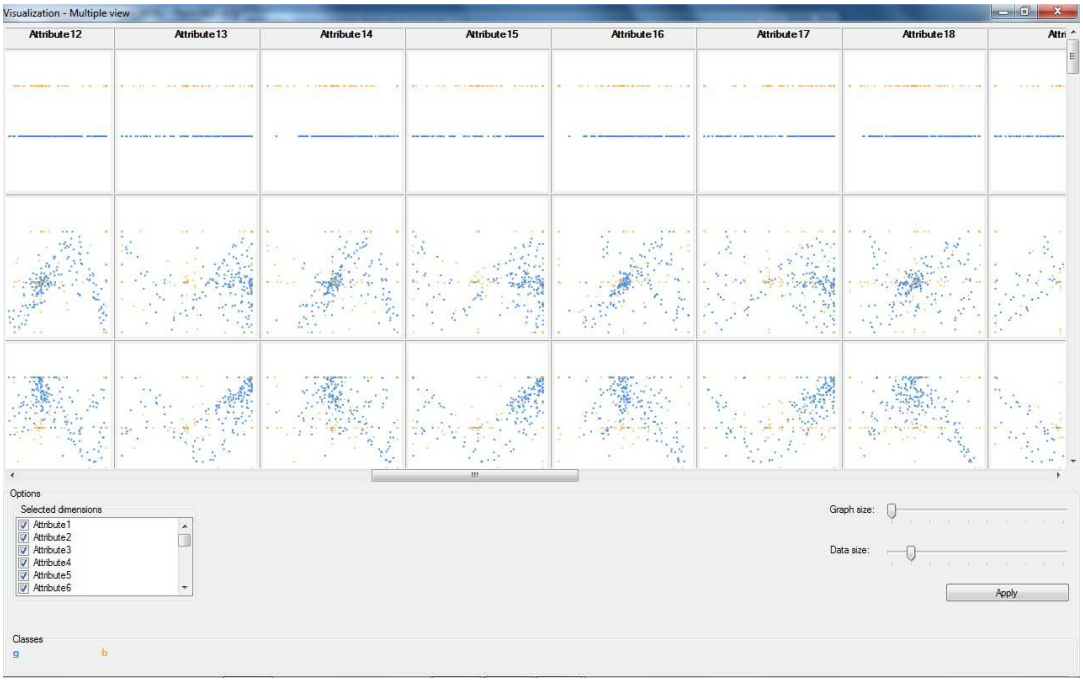


Figure 4.4.2.1 – Graph Table

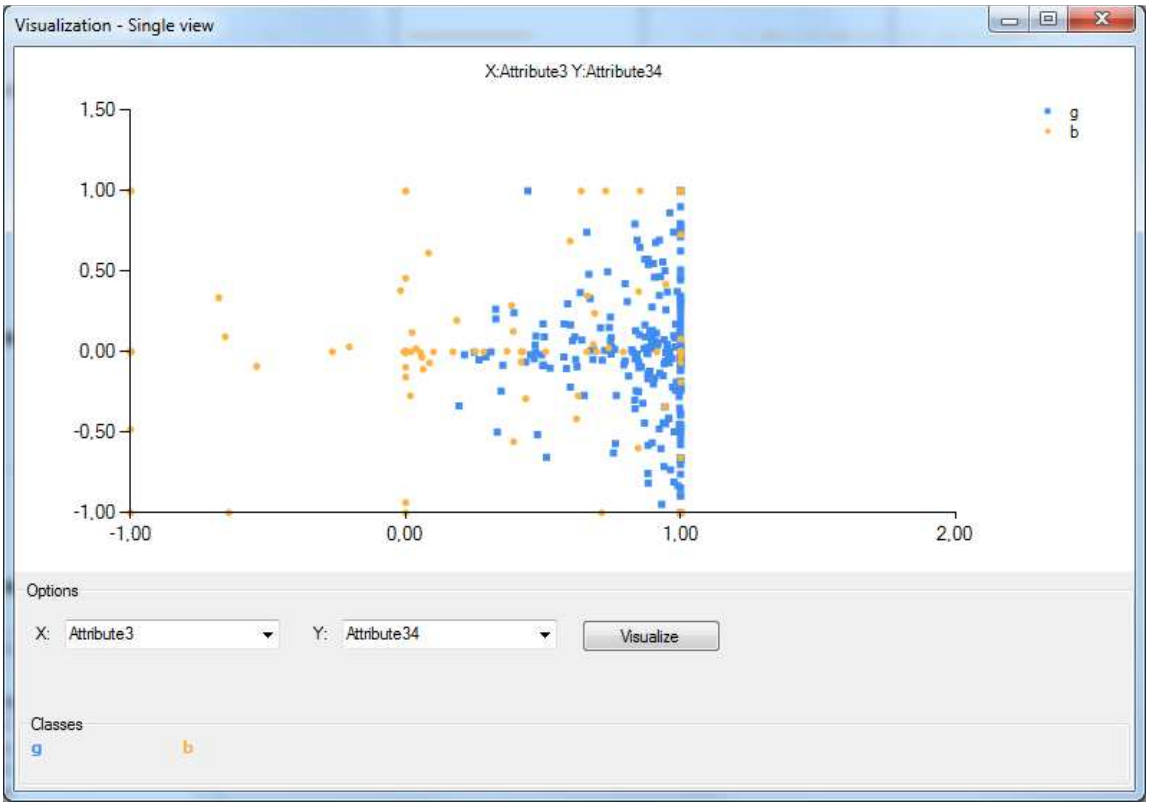
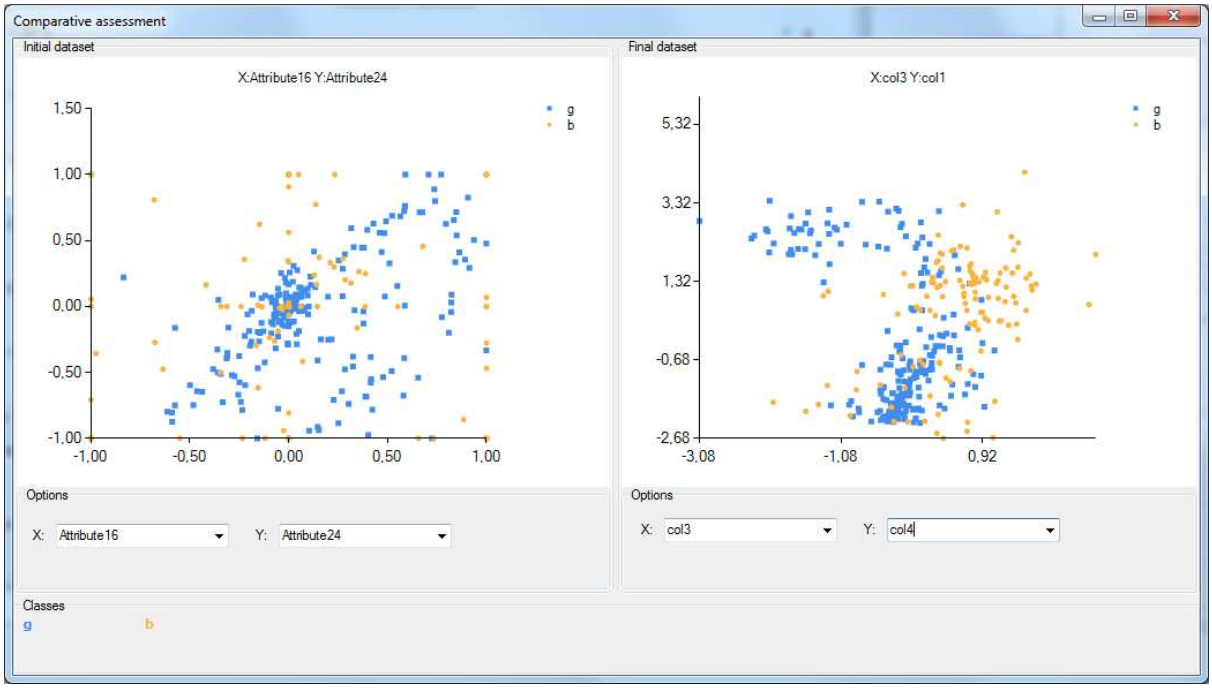


Figure 4.4.2.2. – Data Point Graph



Εικόνα 4.4.2.3 – Comparative Data Point Graph

4.4.3. "Comparison" projection

The last available projection is comparison. With this projection you can compare contrastively two columns graphs. On the left graph you can select any dimension from the initial database. On the right graph you can choose any dimension from the reduced database. One such visualization is presented in figure 4.4.3.1.

Εικόνα 4.4.3.1 – Comparison window

4.5. Fifth level - Data mining with Weka tool

The last level of the application is data mining through Weka tool. Weka is an open source software, that allows data mining through the execution of certain algorithms and processes. Clustering, classification and association algorithms will be used for xSDR application.

Proceeding onto the data visualization level, you reach the installed Weka modules management screen. This screen is depicted in figure 4.5.1. The management window has three areas: on the left you can find a list of all the available algorithms, divided in categories according the algorithm type; on the right upper part of the screen you can see the Weka command, that is going to be executed. If you have a certain familiarity with the command you can type it directly.

Otherwise, you can press the button with the three points, just next to the Weka command bar. This will make appear/produce a window with the possible configurations of the selected algorithm. In figure 4.5.2 you see the equivalent window for the configuration of the SimpleKMeans algorithm. At first, the default values appear, which the user can change at will. Finally, on the lower right part of the screen the results from the algorithm execution appear. By pressing the detail button, you access the results in a new window in text format.

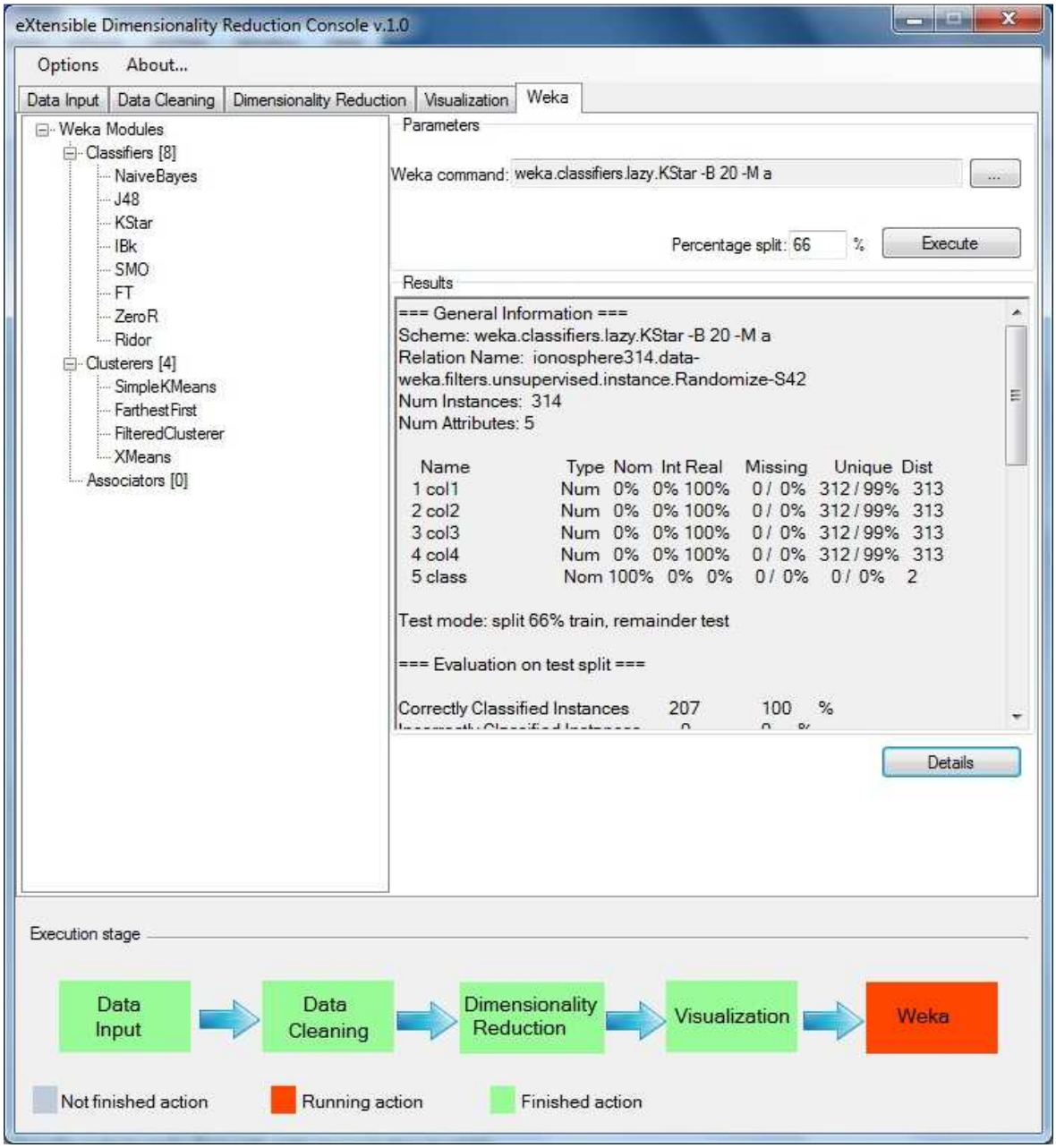
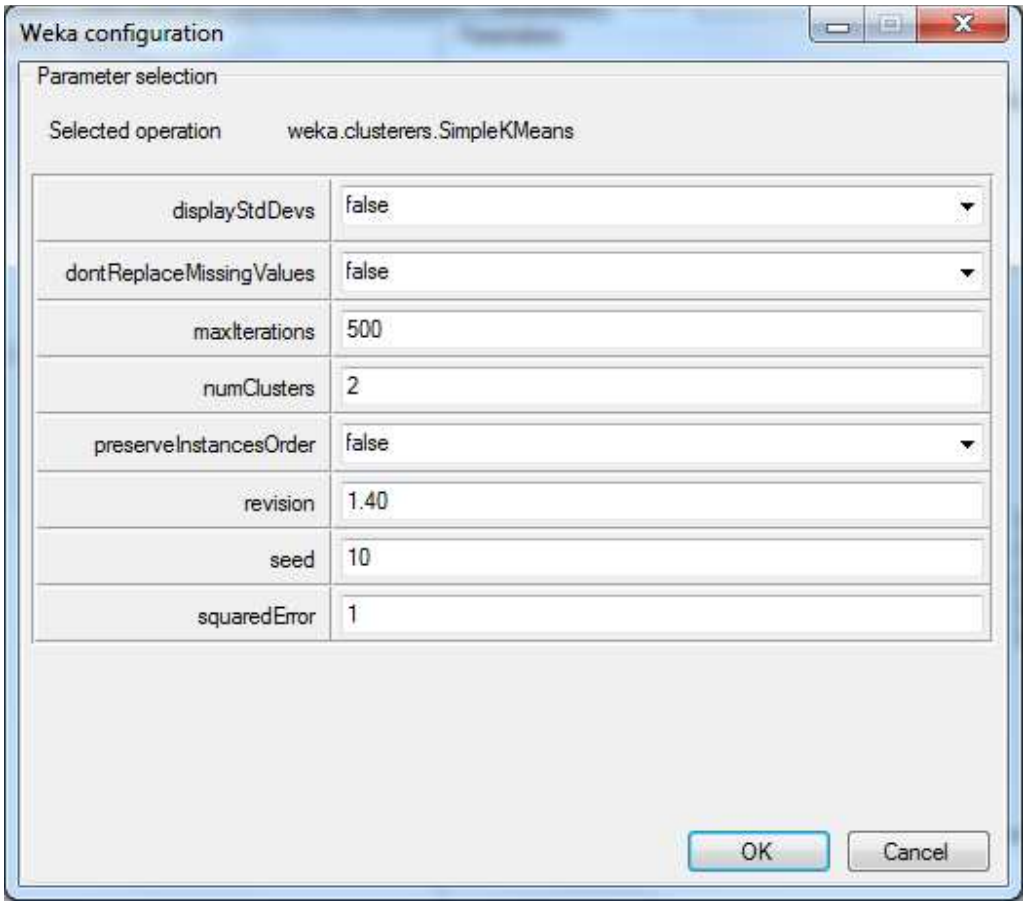


Figure 4.5.1 – Data mining level with the use of Weka tool



Εικόνα 4.5.2. – Configuration of SimpleKMeans algorithm

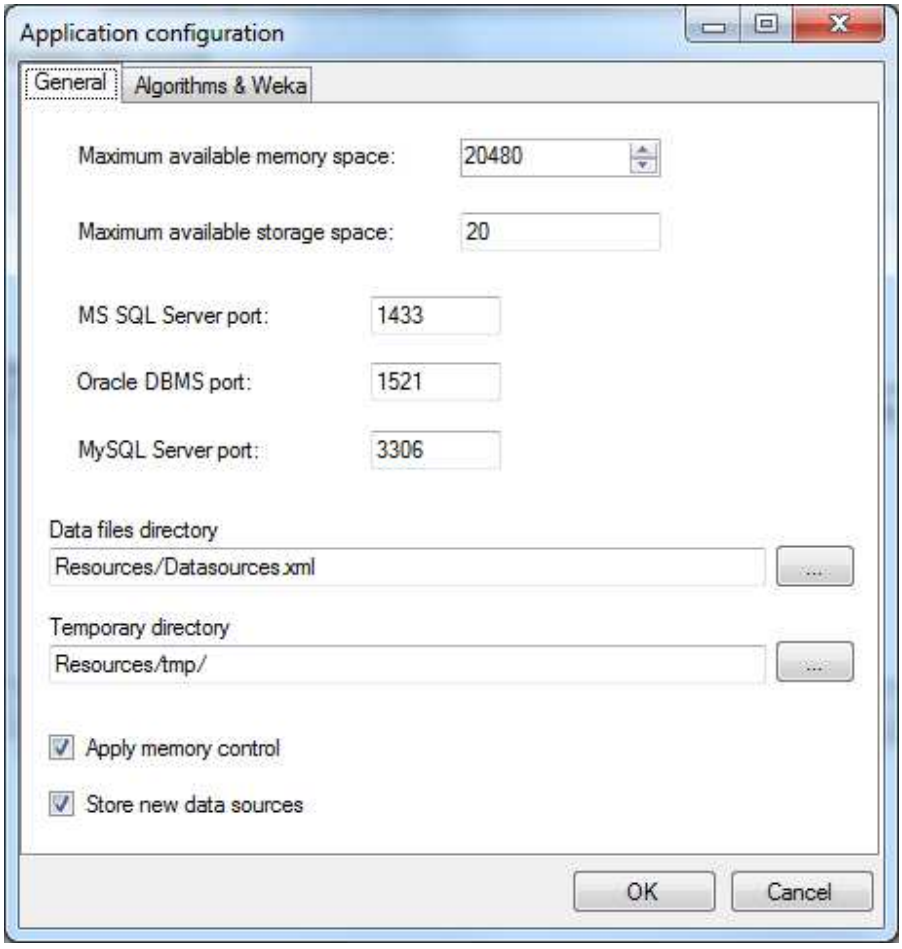
4.6. Application general settings

Through the application main screen menu you can change the basic program settings. From the main screen of the application you go to the settings menu, by pressing the "Options" button and the "Configuration" one. The screen you see is depicted in figure 4.6.1.

In this tab (General) we can define the main settings of the program. In detail, the potential settings are the following:

- Maximum available memory space: it is the maximum memory that is going to be used to load the selected database.
- Maximum available storage space: it is the maximum database number we can store in the memory. In general, there is no limitation to this number; however, we suggest, for performance and system stability reasons, that this number does not exceed 20.
- MS SQL Server Port: The communication port with Microsoft SQL Server (the communication ports are set at the default ports of each base. If you have not set manually another port, leave these settings unchanged).

- Data files Directory: The file in which the xml file that contains the stored database sources is stored.
- Temporary Directory: The file in which the temporary files, that the application created during its execution, are stored.
- Apply memory control: With this option we allow the program to control the memory sufficiency for the algorithm execution.
- Store new data sources: With this option activated, we allow the application to store the new database sources that we create.



Εικόνα 4.6.1 – Application general settings

If you proceed to the next tab (Algorithms & Weka) you will see the available settings, regarding the algorithms and the Weka data mining application.

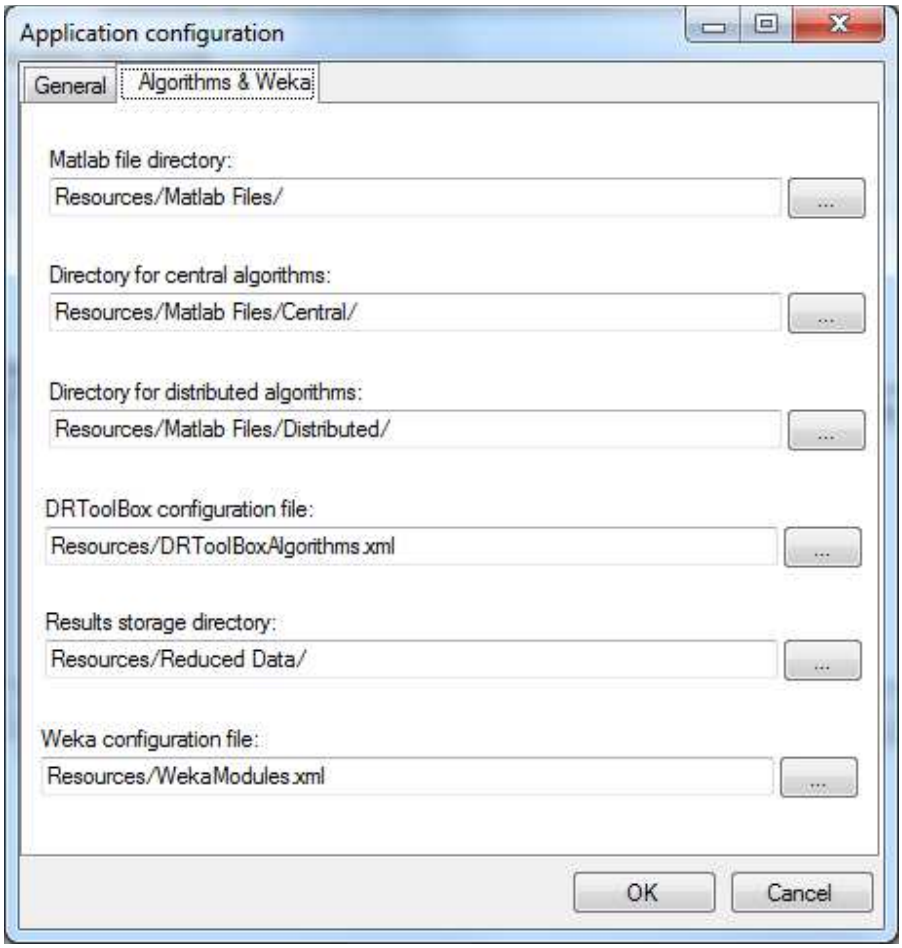


Figure 4.6.2 – Special algorithm and Weka settings

- Matlab file directory: The file in which Matlab files are stored.
- Directory for central algorithms: The file in which the central algorithms we can execute are stored.
- Directory for distributed algorithms: The file in which the available distributed algorithms are stored.
- DRToolbox configuration file: The path for the DRToolbox settings archive.
- Results storage directory: The file in which the dimensionality reduction results are stored.
- Weka configuration file: The file that contains the Weka modules used by the application.

5. Annex

5.1. Links

In this unit you can find the links of the necessary tools for the application execution. xSDR application is designed around the possibility of execution in any modern machine and without the purchase of any additional being necessary.

Microsoft Chart Controls for Microsoft .NET Framework 3.5

http://www.microsoft.com/downloads/details.aspx?FamilyId=130F7986-BF49-4FE5-9CA8-910AE6EA442C&displaylang=en

Microsoft .NET Framework 3.5 with SP1

http://www.microsoft.com/downloads/details.aspx?FamilyId=333325FD-AE52-4E35-B531-508D977D32A6&displaylang=en

Matlab Compiler

According to the Matlab Compiler Licence Agreement, we don't have the rights to give direct links for the compiler. However, we are authorized to distribute, along with our tool, the version of Matlab Compiler Runtime Installer we built during the development phase. So, the required version of MCRInstaller is available through the site of xSDR.

We would also like to notice that every user can ask a free latest version of the compiler through the Mathworks site, using the link provided below:

http://www.mathworks.com/products/compiler/

The last link is not an application but it is very important for the project. It's the Toolbox for Dimensionality Reduction using Matlab.

http://ict.ewi.tudelft.nl/~lvandermaaten/Matlab_Toolbox_for_Dimensionality_Reduction.html

Through this link you can find any new versions of the toolbox and build new assembly files to add them to the tool, as we discussed in an earlier unit.